Computación y Sistemas Vol. 3 No. 4 pp. 264 - 273
 ISSN 1405-5546 Impreso en México

Métricas para la Evaluación de la Complejidad de Bases de Batos Relacionales

Coral Calero, Mario Piattini, Macario Polo, Francisco Ruiz
Grupo ALARCOS
Departamento de Informática. Universidad de Castilla-La Mancha
Ronda de Calatrava, 5. 13071, Ciudad Real (España)
e-mail: {ccalero, mpiattin, mpolo, fruiz}@inf-cr.uclm.es

Artículo recibido el 9 de marzo, 1999; aceptado el 5 de abril, 2000

Resumen

Los datos suponen una parte fundamental de los sistemas de información. Conseguir controlar la complejidad aportada por ellos resulta, por tanto, imprescindible. Una forma de realizar este control es mediante la utilización de métricas específicas para bases de datos, campo este bastante descuidado dentro de la ingeniería del software. Pero no solamente se trata de proponer métricas, estas deben ser sometidas a procesos de verificación formal y de validación empírica con el fin de poder probar sus características y utilidad. Este artículo propone distintas métricas para medir la complejidad de las bases de datos relacionales, caracterizándolas en el marco formal de (Zuse, 1998).

Palabras clave: Métricas, complejidad, bases de datos relacionales

1 Introducción

Es importante que los productos software sean evaluados para cada factor de calidad relevante utilizando métricas validadas o ampliamente aceptadas (ISO, 1994). Uno de estos factores es la mantenibilidad, que representa el mayor problema del desarrollo software ya que supone entre el 60 y el 80 por ciento de los costes del ciclo de vida (Card y Glass, 1990; Pigoski, 1997). Está reconocido que las métricas del software son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento (Briand et al., 1996). Pero las medidas no solamente se utilizan para entender, controlar y probar, sino que también pueden ser utilizadas para que los prácticos e investigadores puedan tomar las mejores decisiones (Pfleeger, 1997).

Los ingenieros de software han propuesto grandes cantidades de métricas para productos, procesos y recursos software (Melton, 1996; Fenton y Pfleeger, 1997). Sin embargo, desde que McCabe propusiera su complejidad ciclomática (McCabe, 1976) hasta ahora, la gran mayoría de las métricas estaban enfocadas a las características del programa dejando a un lado las bases de datos.

Diseñar métricas para medir los datos es importante debido a la relevancia que tanto el tamaño de los datos como su naturaleza tienen en aspectos como el esfuerzo del desarrollo. Por tanto, medir datos puede ayudar a controlar y predecir aspectos del modelo de datos durante el proceso de desarrollo software (MacDonell *et al.*, 1997).

Sneed y Foshag (1998) señalaron recientemente que las métricas para bases de datos habían sido descuidadas, lo cual puede explicarse si tenemos en cuenta que, hasta hace poco, las bases de datos se consideraban como algo secundario sin apenas repercusión en la complejidad total del sistema. Sin embargo, las bases de datos se han constituido en una parte esencial, pasando a ser el núcleo de los sistemas de información, debido a lo cual pensamos que resulta fundamental su medición.

Tres son los factores que influyen en la mantenibilidad: entendibilidad, modificabilidad y testeabilidad, los cuales a su vez están influenciados por la complejidad (Li y Cheng, 1987). Sin embargo, una métrica general para medir la complejidad es como el "santo grial" (Fenton, 1994). Henderson-Sellers (1996) divide la complejidad en tres: computacional, psicológica y representacional, y para la psicológica define tres componentes: complejidad del problema, factores cognitivos humanos y complejidad del producto, en esta última hemos centrado nuestro trabajo.

En este artículo se presentan diferentes métricas para medir la complejidad del esquema de bases de datos relacionales: NA, DRT, RD y NR describiéndolas formalmente siguiendo la teoría de la medida y, concretamente, mediante el marco formal propuesto por Zuse (1998).

En la sección 2 introducimos las métricas propuestas para continuar en la sección 3 haciendo una breve descripción del marco formal de Zuse (1998). En la sección cuarta realizamos la caracterización formal de las métricas propuestas y para terminar, en la sección quinta, presentamos las conclusiones y las futuras líneas de investigación.

2 Métricas propuestas para bases de datos relacionales

Desde que a finales de los sesenta el Dr. Codd propusiera su modelo relacional (Codd, 1970), se ha intensificado la investigación en el campo de las bases de datos y los productos de bases de datos relacionales han generado una importante industria.

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)".

El único indicador utilizado para medir la calidad de una base de datos relacional fue la teoría de la normalización, a partir de la cual Gray et al. (1991) proponen un ratio de normalidad.

En este artículo, proponemos cinco métricas que se pueden clasificar en dos categorías: métricas orientadas a tablas y métricas orientadas a esquemas.

Dentro de las orientadas a tablas proponemos:

Número de atributos (NA)
NA es el número de atributos en todas las tablas del esquema y NA(A) es el número de atributos de la tabla A.

Grado de Referenciabilidad (Referential Degree RD)
RD se define como el número de claves ajenas del esquema relacional y RD(A) es el número de claves ajenas de la tabla A.

Dentro de las orientadas a esquema proponemos:

Profundidad del árbol referencial (Depth Referential Tree DRT)
DRT se define como la longitud del máximo camino referencial del esquema relacional. Los ciclos sólo se consideran una vez.

Ratio de normalidad (Normality ratio NR)
NR se define como el número de tablas en tercera forma normal o superior dividido entre el número de tablas en el esquema:

$$NR = \frac{NT3NF}{NTS}$$

.donde: NT3NF es el número de tablas en 3NF (o superior)
NTS es el número de tablas en el esquema

Aplicamos las métricas definidas al ejemplo de la tabla 1 tomado de Elmasri y Navathe (1997):

```
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),
	NSSSUPER    CHAR(9),
	ND          INT                    NOT NULL,
  CONSTRAINT CLPEMP
	PRIMARY KEY (NSS),
  CONSTRAINT CLESUPEREMP
	FOREIGN KEY (NSSSUPER) 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 CLEGTEDEPTO
	FOREIGN KEY (NSSGTE) REFERENCES EMPLEADO(NSS)
	            ON DELETE SET DEFAULT ON UPDATE CASCADE);

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 (NUMEROP),
  UNIQUE (NOMBREP),
  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 (NUMEROP));

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));
```

Tabla Ejemplo

En la figura 1 se presenta el grafo relacional para el ejemplo anterior, donde las flechas indican las relaciones de integridad referencial:

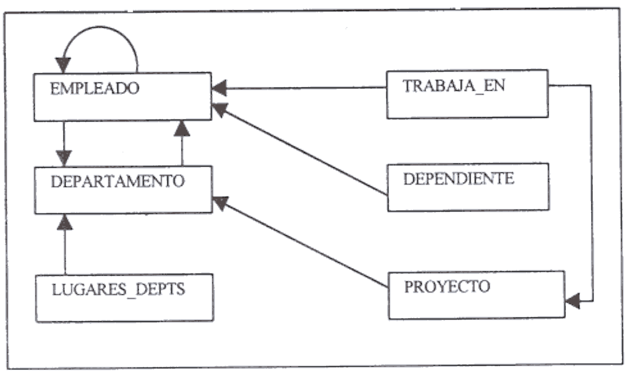


Figura Esquema del ejemplo

Los valores de las métricas correspondientes al ejemplo se muestran en la tabla 2.

	NA	*RD*	*DRT*	*NR*
EMPLEADO	10	2		
DEPARTAMENTO	4	1		
LUGARES_DEPTS	2	1		
PROYECTO	4	1		
TRABAJA_EN	3	2		
DEPENDIENTE	5	1		
ESQUEMA	**28**	**8**	**5**	**1**

Tabla 2. Valores de las métricas para el ejemplo

En la tabla, podemos observar todos los valores obtenidos para las métricas a partir de la información que nos da tanto el ejemplo como su esquema. Por ejemplo, el valor para la métrica DRT es cinco, que se obtiene del camino referencial siguiente:

TRABAJA_EN → PROYECTO → DEPARTAMENTO → EMPLEADO → EMPLEADO → DEPARTAMENTO.
En este punto conviene recordar que, por definición, nuestra métrica DRT sólo considera los ciclos una vez.

3 Marco formal de Zuse

Varios han sido los autores que han propuesto marcos para caracterizar las métricas (Briand et al., 1996; Fenton, 1994; Weyuker, 1998). En este artículo, seguimos el marco formal propuesto por Zuse (1998) para describir las propiedades de las métricas propuestas anteriormente. Este marco es una extensión de la teoría de la medida clásica, en el que se da tanto una base para validar las métricas del software como criterios sobre las escalas de las medidas.

Zuse (1998) describe la medida como algo "necesario ya que los humanos no son capaces de tomar decisiones o enjuiciar de forma clara y objetiva". Medir es mas que producir números, es la combinación de entidades empíricas con entidades numéricas.

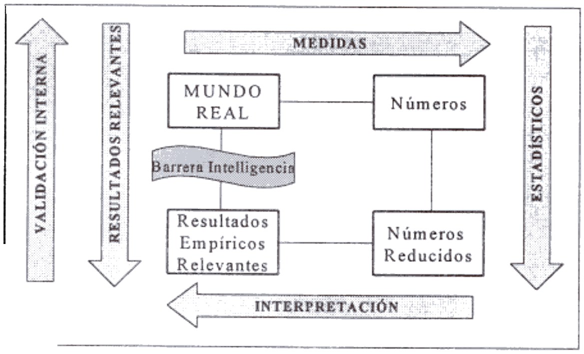


Figura 2.- Proceso de medida según Zuse (1998)

En la figura 2 se muestra el proceso a seguir cuando se mide. Este proceso comienza en el mundo real, el cual contiene los objetos que hay que medir. El interés se centra en establecer "relaciones empíricas" entre objetos, del tipo "mayor que" o "mayor igual que". Estas relaciones empíricas se representan mediante los símbolos "• >" y "• >=" respectivamente. Se representa el Sistema Relacional Empírico como:

A = (A, • >=, o)

Donde A es un conjunto no vacío de objetos, • >= es una relación empírica en A y o es una operación binaria cerrada concatenación) en A

En muchos casos, resulta imposible producir directamente resultados empíricos relevantes debido a la dificultad intrínseca del problema. Existe una barrera de la inteligencia que impide reducir la información sin ayuda. Con las matemáticas y la estadística esta barrera de la inteligencia se puede soslayar. Debemos, por tanto, traducir los objetos empíricos con sus relaciones a objetos numéricos con sus relaciones. Para ello definimos un homomorfismo para realizar el mapeo entre la información empírica y la numérica. El Sistema Numérico Relacional puede ser definido como: B = ($\Re$, >=, +), donde $\Re$ son los números reales, >= una relación en $\Re$, y + una operación binaria en $\Re$.

Por tanto una medida va a ser un mapeo u: A -> $\Re$ tal que para todo a, b pertenecientes al mundo real A se cumple que:

a • >= b $\Leftrightarrow$ u(a) > = u(b)

Una vez establecido el mapeo, se utilizan las matemáticas y la estadística para procesar la información calculando, por ejemplo, las medias o las varianzas.

Tal como señala Zuse (1998), las operaciones de concatenación (o) resultan confusas en el área de la ingeniería del software, aunque nos permiten definir estructuras de medidas más potentes, dando a los números una interpretación mucho más precisa. Además, no es necesario combinar los objetos con una operación de concatenación existente, basta con tener A1, A2, A1 o A2 $\in$ A y una f(A1, A2): A x A -> A. Así, una regla de combinación se puede definir como:

u(A1 o A2) = f(u(A1), u(A2)).

Zuse define un conjunto de propiedades para las medidas las cuales caracterizan diferentes estructuras de medida. Las más importantes son las siguientes:

3.1 Estructura Extensiva Modificada

Axioma1: (A, • >=) (orden débil)
Axioma2: A1 o A2 • >= A1 (positividad)
Axioma3: A1 o (A2 o A3) $\approx$ (A1 o A2) o A3 (asociatividad débil)
Axioma4: A1 o A2 $\approx$ A2 o A1 (conmutatividad débil)
Axioma5: A1 • >= A2 $\Rightarrow$ A1 o A • >= A2 o A (monotonicidad débil)
Axioma6: Si A3 • > A4 entonces para cualquier A1, A2, existe un número natural n, tal que A1o nA3 • >A2 o nA4 (axioma arquimedeano)
Sabiendo que una relación binaria • >= es de orden débil si es transitiva y completa:
A1 • >= A2, y A2 • >= A3 $\Rightarrow$ A1 • >= A3
A1 • >= A2 o A2 • >= A1

3.2. Condiciones de Independencia

Las condiciones de independencia son cumplidas por las estructuras extensivas y son prerrequisito de la existencia de una regla de combinación.

C1: $A1 \approx A2 \Rightarrow A1 \text{ o } A \approx A2 \text{ o } A$ y $A1 \approx A2 \Rightarrow A \text{ o } A1 \approx A \text{ o } A2$
C2: $A1 \approx A2 \Leftrightarrow A1 \text{ o } A \approx A2 \text{ o } A$ y $A1 \approx A2 \Leftrightarrow A \text{ o } A1 \approx A \text{ o } A2$
C3: $A1 \bullet\!>= A2 \Rightarrow A1 \text{ o } A \bullet\!>= A2 \text{ o } A$, y $A1 \bullet\!>= A2 \Rightarrow A \text{ o } A1 \bullet\!>= A \text{ o } A2$
C4: $A1 \bullet\!>= A2 \Leftrightarrow A1 \text{ o } A \bullet\!>= A2 \text{ o } A$, y $A1 \bullet\!>= A2 \Leftrightarrow A \text{ o } A1 \bullet\!>= A \text{ o } A2$
Dónde $A1 \approx A2$ si y sólo si $A1 \bullet\!>= A2$ y $A2 \bullet\!>= A1$, y $A1 \bullet\!> A2$ si y sólo si
$A1 \bullet\!>= A2$ y no ($A2 \bullet\!>= A1$).

3.3 Relación Modificada de Creencia (*belief*)

Zuse incorpora esta nueva estructura para caracterizar las métricas orientadas a objetos. Este tipo de métricas suelen no cumplir ni la estructura modificada extensiva ni las condiciones de independencia. En este caso, la medida es u: $\Im \rightarrow \Re$, dónde $\Im$ es el conjunto de subconjuntos finitos de un conjunto enumerable X.

La forma de actuar si la métrica no cumple la estructura extensiva, será comprobar si cumple la función modificada de crédito: $u(A \cup B) \geq u(A) + u(B) - u(A \cap B)$

Si la cumple ya podremos caracterizar la métrica como una estructura de crédito, si no la cumple habrá que ir comprobando uno a uno los axiomas de la estructura de crédito modificada:

MRB1: $\forall A, B \in \Im: A \bullet\!>= B \text{ o } B \bullet\!>= A$ (compleción)
MRB2: $\forall A, B, C \in \Im: A \bullet\!>= B$ y $B \bullet\!>= C \Rightarrow A \bullet\!>= C$ (transitividad)
MRB3: $\forall A \supseteq B \Rightarrow A \bullet\!>= B$ (axioma de dominancia)
MRB4: $\forall (A \supset B, A \cap C = \phi) \Rightarrow (A \bullet\!>= B \Rightarrow A \cup C \bullet\!> B \cup C)$ (monotonicidad parcial)
MRB5: $\forall A \in \Im: A \bullet\!>= 0$ (positividad)

Es importante advertir que cuando una métrica cumple el axioma de orden débil de la estructura extensiva modificada, también cumple los axiomas de compleción y de transitividad de la estructura de creencia (*belief*).

Por último, debemos recordar que existen cinco tipos de escalas, que, de forma jerárquica, son: nominal, ordinal, intervalo, ratio y absoluta. Cada tipo de escala se define por transformaciones admisibles. Las medidas del software arrancan en la escala ordinal (Zuse, 1998).

Seguidamente, se presenta la transformación que del marco formal se ha hecho para adaptarlo a las bases de datos relacionales, comprobando si las métricas propuestas por los autores verifican los distintos axiomas.

4 Caracterización de las métricas de complejidad para bases de datos relacionales

En los sistemas de bases de datos relacionales, y para nuestros propósitos, el Sistema Relacional Empírico se puede definir como:

$$\mathbf{R} = (R, \bullet\!>=, \text{ o})$$

Donde R es un conjunto no vacío de relaciones (tablas), $\bullet\!>=$ es la relación empírica "mas o igual complejo que" en R y o es una operación binaria (concatenación) cerrada en R.

En nuestro caso, la operación de concatenación elegida dependerá del tipo de métrica con la que estemos trabajando:

Si trabajamos con una de las métricas orientada a tablas, escogeremos la combinación natural como operación de concatenación. Esta combinación natural se define de forma general como (Elmasri y Navathe, 1997) :

$$Q \leftarrow R_{(<lista1>^{*},<lista2>)} S$$

Donde <lista1> especifica una lista de i atributos de R y <lista2> es una lista de i atributos de S. Estas listas son utilizadas para realizar las condiciones de comparación de igualdad entre pares de atributos. Después, estas condiciones son relacionadas mediante el operador AND. Solamente la lista correspondiente a la relación R se preserva en Q.

Hay que tener en cuenta que la combinación natural, si las tablas que se combinan no tienen columnas en común, deriva en el producto cartesiano. Además, la combinación natural se puede realizar a través de una clave ajena y su clave principal o a través de dos columnas cualesquiera definidas sobre el mismo dominio.

Así pues habrá que tener todo esto en cuenta a la hora de diseñar las operaciones de combinación de cada una de las métricas orientadas a tablas.

Si trabajamos con las métricas orientadas a esquemas, la operación de concatenación se limitará a la integración de los dos esquemas origen, preservando todas las características originales de ambos en un nuevo esquema.

4.1 Métricas orientadas a tabla

NA, número de atributos

La métrica NA es un mapeo: NA: R -> $\Re$ tal que se cumple para todas las relaciones Ri y Rj $\in$ R:
Ri $\bullet$ >= Rj $\Leftrightarrow$ NA(Ri) >= NA(Rj).
Su regla de combinación se define como:

NA (Ri o Rj) = NA (Ri) + NA(Rj) – NA(Ri $\cap$ Rj)

Dónde NA(Ri $\cap$ Rj) es el número de atributos comunes a (pertenecen a la intersección[1] de) Ri y Rj.

En Calero y Piattini (1999) analizamos a fondo las características formales de esta métrica, demostrando que NA no asume una estructura extensiva ni cumple las condiciones de independencia, pero que, sin embargo, si cumple la estructura de creencia (*belief*).

Por tanto, podemos caracterizar NA como una métrica sobre el nivel de la escala ordinal que asume la relación modificada de creencia (*belief*).

Métrica RD

Teniendo en cuenta que la combinación natural entre tablas afecta al número de claves ajenas, podemos definir la regla de combinación para esta métrica como:

$$RD(R_i o R_j) = RD(R_i) + RD(R_j) - v$$

Dónde v es una variable que puede tomar valores 0 ó

En Calero y Piattini (1999) analizamos a fondo las características formales de esta métrica, demostrando que RD no asume una estructura extensiva ni cumple las condiciones de independencia, pero que, sin embargo, si cumple la estructura de creencia (*belief*).

Por tanto, podemos caracterizar RD como una métrica sobre el nivel de la escala ordinal que asume la relación modificada de creencia (*belief*).

[1] Esta intersección es distinta de la operación intersección entre dos relaciones del álgebra relacional.

4.2. Métricas orientadas a esquema

Métrica DRT

La métrica DRT es un mapeo: DRT: R -> $\Re$ tal que se cumple para todas las relaciones Ri y Rj $\in$ R: Ri $\bullet$ >= Rj $\Leftrightarrow$ DRT(Ri) >= DRT(Rj).

Teniendo en cuenta la operación de concatenación definida para las métricas orientadas a esquema, podemos concluir que la regla de combinación para esta métrica será:

DRT(Ri o Rj)=max (DRT(Ri) , DRT(Rj))

Pasamos ahora a comprobar uno por uno todos los axiomas de la estructura extensiva modificada para ver si la métrica las cumple: Sean R1, R2 y R3 tres esquemas de bases de datos relacionales, resulta obvio que: DRT(R1)>=DRT(R2) o DRT(R2)>=DRT(R1) y también que DRT(R1)>=DRT(R2) y DRT(R2)>=DRT(R3) $\Rightarrow$ DRT(R1)>=DRT(R3), por tanto DRT cumple el axioma de orden débil (axioma 1). El axioma de positividad (axioma 2) también se va a cumplir ya que al integrar dos esquemas, el valor de la métrica en el esquema resultante será el máximo de los valores para los esquemas iniciales y, por tanto, el valor resultante nunca podrá ser menor. La asociativa (axioma 3) también se cumple ya que en cualquier caso obtenemos para la métrica el valor máximo de los valores de los esquemas originales. Lo mismo pasa con la conmutativa (axioma 4) ya que en ambos casos obtendremos el mismo esquema resultante. Respecto al axioma 5 (monotonicidad débil) también se va a cumplir ya que siempre trabajamos con máximos. El axioma arquimedeano (axioma 6) no se va a cumplir ya que resulta evidente que la métrica es idempotente (el valor de la métrica para sucesivas concatenaciones de un esquema consigo mismo siempre va a ser igual).

Por tanto, la métrica no es una estructura extensiva modificada.

Pasamos ahora a comprobar las condiciones de independencia. La condición primera se va a cumplir ya que si partimos del mismo valor de los dos esquemas originales e integramos el mismo esquema a ambos, obtenemos que ambas integraciones tendrán el mismo valor para la métrica. El mismo tipo de razonamiento nos sirve para comprobar que la condición segunda no tiene porque darse ya que puede que el valor de la métrica para (A1oA) y el de (A2oA) venga dado por el hecho de que A tenga un valor mayor para la métrica que A1 o A2, por lo que estos dos esquemas pueden tener distintos valores para la métrica, no cumpliéndose por tanto la condición de independencia segunda. Por cumplir el axioma quinto de la estructura extensiva, también va a cumplir las condiciones tercera y cuarta.Por tanto, DRT no cumple las condiciones de independencia.

Por último, comprobamos la estructura de creencia (*belief*). Como la métrica cumple el axioma de orden débil también va a cumplir los axiomas uno y dos de la estructura de creencia (*belief*). El tercer axioma también se cumple ya que si todos los caminos referenciales de un esquema B están incluidos en otro esquema A, el valor de la métrica para el esquema A será mayor o igual que el valor de la métrica para el esquema B. El axioma de monotonicidad parcial (axioma 4) no se va a cumplir ya que tanto a A como a B se le van a añadir los mismos caminos referenciales (de C), y si el valor de la métrica para C es mayor que para A, después de la integración (de A con C y de B con C) tendremos una relación de igualdad en el valor de la métrica para los esquemas resultantes (DRT(AoC)=DRT(C)=DRT(BoC)). Finalmente, el axioma de la positividad (axioma 5) si se va a cumplir ya que la longitud de un camino referencial es siempre mayor o igual que cero.

Es decir, DRT no asume los axiomas de la estructura de creencia (*belief*).

Por tanto, podemos caracterizar DRT como una métrica en el nivel de la escala ordinal.

Métrica NR

Lo primero que debemos hacer es definir la regla de combinación de la métrica. Cuando combinamos esquemas, según la operación de combinación que hemos definido, siempre se va a cumplir que tanto el número de tablas en tercera forma normal como el número de tablas totales del esquema resultante van a ser la suma del número de tablas en tercera forma normal y la suma del número de tablas de los dos esquemas de partida.

Teniendo todo lo anterior en cuenta, podemos definir la regla de combinación como:

$$NR(R_i o R_j) = \frac{NT3FN_i + NT3FN_j}{}$$

Comenzamos comprobando que no asume una estructura extensiva modificada. Sean R1, R2 y R3 tres esquemas relacionales, siempre se va a cumplir que NR(R1)>=NR(R2) o NR(R2)>=NR(R1), y además se va a cumplir que si NR(R1)>=NR(R2) y NR(R2)>=NR(R3) $\Rightarrow$ NR(R1)>=NR(R3) por lo que la métrica cumple el axioma de orden débil (axioma1). El axioma de la positividad (axioma 2) no se cumple ya que si, por ejemplo, en el esquema A1 tenemos que todas las tablas están en tercera forma normal o superior (su valor para la métrica será 1) y en el esquema A2 tenemos una sola tabla que además no está en tercera forma normal, al combinar ambos esquemas vamos a obtener para el esquema resultante que el divisor aumentará en 1 (respecto al valor de la métrica para A1) pero el numerador no variará por lo que el valor final de la métrica para el esquema resultante disminuirá respecto al de A1.La asociatividad (axioma 3) y la conmutatividad (axioma 4) también van a cumplirse debido al tipo de combinación definida para los esquemas. El axioma de la monotonicidad débil (axioma 5) se va a cumplir ya que si partimos de dos esquemas con unos determinados valores para las métricas, y añadimos a ambos el mismo esquema, es decir el mismo número de tablas en tercera forma normal y totales, la relación existente entre los valores numéricos se mantendrá. Por último, el axioma 6, el axioma arquimedeano no se cumple ya que la métrica es idempotente (sucesivas integraciones de un esquema consigo mismo no van a variar el valor para la métrica).

Así pues podemos concluir que la métrica NR no es una estructura extensiva modificada.

Veamos ahora las condiciones de independencia. La primera condición no se cumple, el siguiente ejemplo lo demuestra:

Sean P1, P2 y P3 esquemas de base de datos relacionales de forma que toman los siguientes valores para la métrica NR:

$$NR(P1) = \frac{10}{1000} = \qquad NR(P2) = \frac{50}{5000} =$$

$$NR(P3) = \frac{7}{1000}$$

$$NR(P1oP3) = \frac{10+7}{2000} \qquad NR(P2oP3) = \frac{50+7}{6000}$$

$$NR(P1oP3) \neq NR(P2oP3)$$

Al no cumplir la condición 1, no puede cumplir la 2. Respecto a la tercera condición de independencia se cumple ya que cumplió el axioma 5 (monotonicidad débil). La cuarta condición de independencia también se cumple ya que si A1oA>=A2oA es porque en A1oA hay mayor (o igual) número de tablas en tercera forma normal que en A2oA, como el esquema A es común a ambas concatenaciones, la relación numérica sólo va a poder darse si en A1 hay mayor (o igual) número de tablas en tercera forma normal que en A2.

Como conclusión, podemos afirmar que NR no cumple todas las condiciones de independencia.

Pasemos ahora a estudiar la estructura de creencia (*belief*). Por cumplir el axioma de orden débil, cumple las propiedades uno y dos de la estructura de creencia (*belief*). Respecto a la tercera, si todas las tablas en tercera forma normal de B están en A, el valor de la métrica para A será mayor que para B. Sin embargo, la cuarta propiedad no se cumple. Supongamos el siguiente ejemplo:

A tiene 4 tablas: 2 en tercera forma normal y 2 no (NR(A)=2/4).

B tiene dos tablas (ambas incluidas en A): en tercera forma normal y otra no (NR(B)=1/2).

C tiene 2 tablas ambas en tercera forma normal (NR(C)=2/2).

Si integramos A con C obtenemos 6 tablas, de las cuales 4 están en tercera forma normal (NR(AoC)=4/6).

Si integramos B con C obtenemos 4 tablas de las cuales 3 están en tercera forma normal (NR(BoC)=3/4).

Y como vemos la propiedad no se cumple ya que NR(AoC) no es mayor que NR(BoC).

La quinta si se cumple ya que si no hay ninguna tabla en tercera forma normal la métrica valdrá 0, pero nunca podrá tomar un valor inferior.

Por tanto, la métrica no cumple la estructura de creencia (*belief*).

Como conclusión, sólo podemos caracterizar NR como una métrica en el nivel de la escala ordinal.

En la tabla 3 se resumen los resultados obtenidos para cada métrica respecto de los axiomas propuestos:

Propiedades	NA	RD	DRT	NR
Axioma 1	SI	SI	SI	SI
Axioma 2	NO	NO	SI	NO
Axioma 3	SI	SI	SI	SI
Axioma 4	SI	SI	SI	SI
Axioma 5	NO	NO	SI	SI
Axioma 6	NO	NO	NO	NO
Cond. Ind. 1	NO	NO	SI	NO
Cond. Ind. 2	NO	NO	NO	NO
Cond. Ind. 3	NO	NO	SI	SI
Cond. Ind. 4	NO	NO	SI	SI
MRB 1	SI	SI	SI	SI
MRB 2	SI	SI	SI	SI
MRB 3	SI	SI	SI	SI
MRB4	SI	SI	NO	NO
MRB5	SI	SI	SI	SI

Tabla 3.- Resultados para las métricas

5 Conclusiones

Es necesaria una mayor investigación en los aspectos de la medida del software, tanto desde el punto de vista teórico como práctica (Neil, 1994; Glass, 1996; Basili, 1999).

Pensamos que es muy interesante disponer de métricas para bases de datos relacionales. Estas métricas pueden utilizarse para llamar la atención de los jefes de proyecto sobre determinados esquemas relacionales que tengan unos requisitos muy estrictos de mantenibilidad, de forma parecida a la propuesta por Chidamber et al., (1998) para modelos orientados a objetos.

En este artículo hemos presentado una serie de propuestas para medir la complejidad de bases de datos relacionales, que vaya más allá de la sola teoría de la normalización. Estas métricas se han caracterizado utilizando un conjunto de principios de la teoría de la medida.

Creemos que es también fundamental llevar a cabo una validación empírica para demostrar la utilidad de la medida en la práctica. Una medida puede ser correcta desde el punto de vista de la teoría de la medida (consistente con el sistema relacional empírico) pero irrelevante de cara al problema a resolver. Con este fin se han llevado a cabo algunos experimentos que demuestran la influencia del valor de la métrica RD en la comprensibilidad de esquemas relacionales (Calero et al., 1999a). De todas maneras, somos conscientes de las limitaciones de los experimentos controlados (Basili, 1999) por lo que se han iniciado diversos análisis de bases de datos "reales" en colaboración con organismos y empresas.

Igualmente, sería conveniente poder proporcionar valores límites para las métricas. Con este objetivo, se están estudiando una serie de experimentos y casos reales, can métricas ya validadas, que nos permitan determinar esos valores límites.

También hay que tener en cuenta que la utilidad de las métricas mejoraría mucho si se diera una guía clara de aplicación a lenguajes específicos (Churcher y Shepperd, 1995). Estamos construyendo herramientas para obtener de forma automática estas medidas en entornos que soporten el lenguaje SQL.

Asimismo pensamos complementar estas métricas definidas para bases de datos relacionales con otras para bases de datos activas (Diaz y Piattini, 1999) y bases de datos objeto-relacionales (Piattini et al, 1998; Calero et al., 1999b).

Agradecimientos

Este trabajo forma parte del proyecto MANTICA, parcialmente financiado por la CICYT y la Unión Europea (1FD97-0168).

Los autores quieren expresar su agradecimiento a los revisores anónimos por sus valiosos comentarios. Algunos de ellos ya se han incluido en este artículo y otros se están estudiando para los futuros trabajos.

Referencias

Basili, V. R. (1999). Using experiments to build a body of knowledge.20/1/99. Madrid. España.

Briand, L.C., Differding, C.M. y **Rombach, D.** (1996). Practical Guidelines for Measurement-Based Process Improvement. Software Process-Improvement and Practice 2, 253-280.

Briand, L.C., Morasca, S. y **Basili, V.** (1996). Property-based software engineering measurement. IEEE Transactions on Software Engineering, 22(1): 68-85.

Calero, C., Piattini, M., Polo, M. y **Ruiz, F.** (1999a). Validating referential integrity as a database quality metric. Proc of International Conference on Enterprise Information Systems, Setubal, 27-30 Marzo.

Calero, C., Piattini, M., Ruiz, F. y **Polo, M.** (1999b), Validation of metrics for Object-Relational Databases, *International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (ECOOP99),* (Lisbon ,Portugal. June 1999), 14-18

Calero, C. y **Piattini, M.** (1999). Caracterización formal de métricas para bases de datos relacionales. *IV Jornadas de Ingeniería del Software y Bases de Datos (JISBD99),* Cáceres (España), Noviembre 1999, 139-150

Card, D.N. y **Glass, R.L.** (1990). Measuring Software Design Quality. Englewood Cliffs. USA.

Chidamber, S.R., Darcy, D.P. y **Kemerer, C.** (1998). Managerial Use of Metrics for Object-Oriented Software: An Exploratory Analysis. IEEE Trans. On Software Engineering, Vol. 24, Nº 8, 629-639.

Churcher, N.J. y **Shepperd, M.J.** (1995). Comments en "A Metrics Suite for Object-Oriented Design". IEEE Trans. on Software Engineering, 21(3): 263-265.

Codd, E.F. (1970). A Relational Model of Data for Larged Shared Data Banks. CACM, 13 (6), 377-387.

Date, C.J. (1995). An Introduction to Database Systems. 6th. Addison-Wesley, Reading, Massachusetts.

Díaz, O. y **Piattini. M.** (1999). Metrics for active databases maintainability. Accepted in CAISE'99. Heidelberg, June 16-18.

Elmasri R y **Navathe S.** (1997). Database Systems. Second edition. Addison-Wesley. Massachussets

Fenton, N. (1994). Software Measurement: A Necessary Scientific Basis. IEEE Transactions on Software Engineering, 20(3): 199-206.

Fenton, N. y **Pfleeger,. S. L.** (1997). Software Metrics: A Rigorous Approach 2nd. edition. London, Chapman & Hall.

Glass, R. (1996). The Relationship Between Theory and Practice in Software Engineering. IEEE Software, November, 39 (11), 11-13.

Gray R.H.M., Carey B.N., McGlynn N.A. y **Pengelly A.D.**, (1991), Design metrics for database systems. BT Technology J, Vol 9, 4 Oct, 69-79

Henderson-Sellers, B. (1996). Object-oriented Metrics - Measures of complexity. Prentice-Hall, Upper Saddle River, New Jersey.

ISO, (1994). Software Product Evaluation-Quality Characteristics and Guidelines for their Use. *ISO/IEC Standard 9126*, Geneva.

Li, H.F. y **Cheng, W.K.** (1987). An empirical study of software metrics. IEEE Trans. on Software Engineering, 13 (6): 679-708.

MacDonell, S.G., Shepperd, M.J. y **Sallis, P.J.** (1997). Metrics for Database Systems: An Empirical Study. Proc. Fourth International Software Metrics Symposium – Metrics'97, Albuquerque. IEEE Computer Society, pp. 99-107.

McCabe, T.J. (1976). A complexity measure. IEEE Trans. Software Engineering 2(5): 308-320.

Melton, A. (ed.) (1996). Software Measurement. London, International Thomson Computer Press.

Neil, M. (1994) Measurement as an Alternative to Bureaucracy for the Achievement of Software Quality. Software Quality Journal 3 (2), 65-78.

Piattini, M., Calero, C., Polo, M. y **Ruiz, F.** (1998). Maintainability in Object-Relational Databases. Proc of The European Software Measurement Conference FESMA 98, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (eds.), 223-230.

Pigoski, T.M. (1997). Practical Software Maintenance. Wiley Computer Publishing. New York, USA.

Pfleeger, S. L. (1997). "Assessing Software Measurement". IEEE Software. March/April, pp. 25-26.

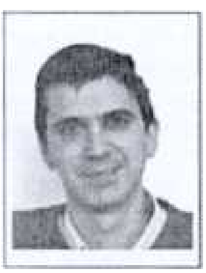
Sneed, H.M. y **Foshag, O.** (1998). Measuring Legacy Database Structures. Proc of The European Software Measurement Conference FESMA 98, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (eds.), 199-211.

Weyuker, E.J. (1988). Evaluating software complexity measures. IEEE Transactions on Software Engineering 14(9):1357-1365.

Zuse, H. (1998). A Framework of Software Measurement. Berlin, Walter de Gruyter.

__Coral Calero__ es Licenciada en Informática por la Universidad de Sevilla. Es Ayudante de Escuela Universitaria en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha en Ciudad Real. Forma parte del grupo de investigación ALARCOS, de esa misma Universidad. En la actualidad está realizando su tesis doctoral sobre Métricas para Bases de Datos Avanzadas. Es autora de varios artículos y conferencias nacionales e internacionales sobre este tema. Pertenece a la Asociación de Técnicos de Informática (ATI) y forma parte del grupo de Calidad de esta misma asociación.

Mario Piattini es Doctor e Ingeniero en Informática por la Universidad Politécnica de Madrid. CISA por la ISACA. Ha sido director del Departamento de Desarrollo de la empresa SiE, y socio-fundador de Cronos Ibérica, S.A. Ha trabajado como consultor y profesor para numerosos organismos y empresas. Actualmente es Profesor Titular de Universidad en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha en Ciudad Real, donde dirige el grupo de investigación Alarcos. Autor de varios libros así como de un centenar de artículos en revistas y conferencias nacionales e internacionales. Pertenece a diversas asociaciones profesionales y es consejero de la Fundación DINTEL.



__Macario Polo Usaola__ es Licenciado en Informática por la Universidad de Sevilla. Trabajó como analista-programador en varias empresas hasta su incorporación a la Universidad de Castilla-La Mancha, en donde trabaja actualmente como Ayudante de Facultad. Además de ser autor de varias novelas, es también autor o coautor de diversas publicaciones científicas relacionadas con el mantenimiento y la medición del software y las bases de datos, áreas que constituyen sus principales temas de investigación.

__Francisco Ruiz__ es profesor titular del Departamento de Informática de la Universidad de Castilla-La Mancha (UCLM) en Ciudad Real, España. Sus temas de investigación actuales incluyen: mantenimiento del software, mejora de procesos software, métricas para bases de datos orientadas a objetos, y metodologías para planificar y gestionar proyectos de ingeniería del software. Ha publicado cuatro libros sobre los temas citados y tiene publicaciones en congresos, conferencias y revistas internacionales. Pertenece a diversas asociaciones científicas y profesionales (ACM, IEEE-CS, ATI, AEC, AENOR).