

Algoritmos de Inserción Segura desde Especificaciones Formales de Restricciones de Integridad

Gonzalo Rojas D.

Universidad de Concepción, Depto. de Ing.Informática y Cs. de la Computación
Concepción, Chile, 4081515
grojas@inf.udec.cl

and

Marcela Varas C.

Universidad de Concepción, Depto. de Ing.Informática y Cs. de la Computación
Concepción, Chile, 4081515
mvaras@inf.udec.cl

Abstract

An approach to face the database integrity maintenance problem is presented. Data insertion algorithms based on integrity constraints formal specifications at conceptual level, are generated, using CCER data model [10]. A multi-level database system architecture is presented, based on this proposal. Design process for algorithms associated with each integrity constraint, and a collaborative execution approach, are described. Finally, an example of this proposal, based on a particular integrity constraint, is shown.

Keywords: Database, Integrity constraints, Conceptual Modelling, Data Insertion Algorithms, Database System Architecture.

Resumen

En el presente artículo se presenta una propuesta para abordar el problema de la conservación de la consistencia de bases de datos, que consiste en la generación de algoritmos de inserción a partir de especificaciones formales de restricciones de integridad a nivel conceptual, utilizando el modelo de datos CCER [10]. Se presenta una arquitectura multinivel para sistemas de bases de datos, basada en esta propuesta. Se describe el proceso de diseño de los algoritmos asociados a cada restricción y una estrategia para la ejecución conjunta de los algoritmos generados. Finalmente, se presenta un ejemplo de esta propuesta basado en una restricción de integridad particular.

Palabras Claves: Bases de Datos, Restricciones de Integridad, Modelamiento Conceptual, Algoritmos de Inserción, Arquitectura de Sistemas de Bases de Datos.

1 Introducción

En el presente artículo, se presenta una propuesta para enfrentar el problema de la preservación de la consistencia de bases de datos, desde el nivel de diseño conceptual. Se plantea la utilización de algoritmos transaccionales (en particular, de ingreso de datos), generados directamente desde especificaciones formales de restricciones de integridad.

A partir de la especificación de cada una de las restricciones expresables por el modelo CCER [10], se obtiene un algoritmo de inserción de datos que obliga a su cumplimiento, cerrando la entrada de posibles inconsistencias. De este modo, se obtiene un conjunto de algoritmos generados a partir del mapeo restricción-algoritmo, conjunto denominado Base de Algoritmos de Inserción Segura para CCER o BAIS-CCER.

La consideración de restricciones de integridad en los algoritmos de inserción de datos permitirá, desde las primeras etapas del diseño, velar por el cumplimiento de aquellas y mantener la consistencia de la base de datos. La portabilidad de los procedimientos transaccionales se verá aumentada, por el origen formal de los algoritmos en que se basan.

Eventuales actualizaciones de los algoritmos obtenidos, producto de modificaciones y depuraciones en los esquemas de datos, implicarán un menor costo que las modificaciones de procedimientos construidos en las etapas postreras del desarrollo de la base de datos.

El artículo que se presenta está dividido en seis secciones. En la sección 2, se describe brevemente la problemática que plantea a los diseñadores de bases de datos la consideración de restricciones de integridad. En la sección siguiente, se expone la propuesta de obtención de algoritmos de inserción segura, que incluye un modelo de arquitectura de bases de datos en la que dichos algoritmos se insertan. La sección 4 describe brevemente el mecanismo de mapeo involucrado y estrategias de ejecución de los algoritmos generados. En la sección 5, se revisan las consideraciones de diseño de un algoritmo asociado a una restricción de integridad particular y un ejemplo de uso. Finalmente, la sección 6 muestra las conclusiones del artículo presentado.

2 El problema del manejo de restricciones de integridad.

2.1 Antecedentes

Para que la información contenida en una base de datos refleje con certeza la realidad representada, uno de los aspectos más importantes a considerar es el cumplimiento de las restricciones de integridad durante la realización de transacciones.

Con los modelos tradicionalmente utilizados en el diseño de bases de datos, como el MER [6] y el Relacional [7], se puede expresar una cantidad muy limitada de restricciones, no permitiendo la especificación formal de muchas otras. Esta carencia de expresividad de los modelos obliga a que estas últimas sólo puedan ser implementadas en etapas tardías del desarrollo, o bien, permite que sean completamente ignoradas en la modificación de los datos, poniendo en riesgo la consistencia de la base de datos.

Diseñando con esta estrategia, resulta difícil especificar procedimientos para la manipulación de la base de datos, de modo que sean portables y que su actualización sea de bajo costo, debido a que las restricciones de integridad que éstos deben considerar son implementadas con un alto grado de dependencia de la plataforma.

2.2 Enfoques adoptados para solución

El problema descrito ha sido abordado desde diversas perspectivas. Las principales labores de investigación han adoptado al menos uno de los siguientes enfoques:

- Enriquecimiento de los mecanismos de consulta y de transacciones a la base de datos ya implementada, dotándolos de mayores capacidades de gestión de inconsistencias, como reportes y reparaciones, de acuerdo a las restricciones de integridad del problema.
- Enriquecimiento de los modelos de datos, lo que permite incluir varios tipos de restricciones de integridad directamente en los esquemas, junto con métodos que velen por su cumplimiento.

La mayoría de los trabajos correspondientes al primer enfoque abordan el problema de preservación de integridad de la base de datos desde un punto de vista reactivo, a través de procedimientos que gestionan inconsistencias introducidas a través de transacciones. Se pueden distinguir mecanismos de reparación de inconsistencias [5], adaptación de consultas [4] o notificación del incumplimiento de restricciones [3]. Es decir, como punto partida de estudio, se asume una base de datos inconsistente o con riesgo permanente de serlo.

La propuesta del presente trabajo se basa en la segunda perspectiva (ver, por ejemplo, [2] y [9]), adoptando un enfoque preventivo que aspira a obtener una base de datos que cumpla las restricciones de integridad desde su poblamiento. Mediante la creación de algoritmos de inserción segura, se fortalecerá el modelo de datos CCER [10], un modelo conceptual de bases de datos con mayor capacidad expresiva de restricciones de integridad que modelos como el MER [6] y sus extensiones [1], [8] y el Relacional [7].

3 Propuesta de solución: Base de Algoritmos de Inserción Segura.

3.1 Descripción de la propuesta

Se propone un mecanismo de mapeo para la generación directa de procedimientos algorítmicos de inserción de datos, que obligue al cumplimiento de las restricciones de integridad expresables a través de los formalismos del modelo CCER.

El mecanismo realiza la asociación de un algoritmo de inserción a cada restricción de integridad del modelo, generando un conjunto de procedimientos denominado *Base de Algoritmos de Inserción Segura para CCER (BAIS-CCER)*. Este conjunto podrá ser utilizado por aplicaciones que manipulen la base de datos, implementado como servicios de inserción.

Los algoritmos generados vigilarán el cumplimiento de sus correspondientes restricciones, antes y durante la inserción de datos, no otorgando al usuario la posibilidad de violarlas. Con esto, se genera una base de datos “consistente por inserción”, es decir, la utilización de estos procedimientos busca asegurar la preservación de la consistencia de la base de datos durante el ingreso de datos.

Para la elaboración de los procedimientos de inserción que conforman la BAIS, se asume que los esquemas de datos que servirán como entrada están correctamente diseñados, es decir, no existen conflictos entre las restricciones de integridad que incluyan. Los procedimientos son expresados en un lenguaje independiente de la plataforma de implementación, para asegurar su portabilidad.

3.2 Arquitectura del Sistema de Bases de Datos

El esquema de la figura 1 muestra la arquitectura del sistema de bases de datos en el cual está inserta la Base de Algoritmos de Inserción Segura:

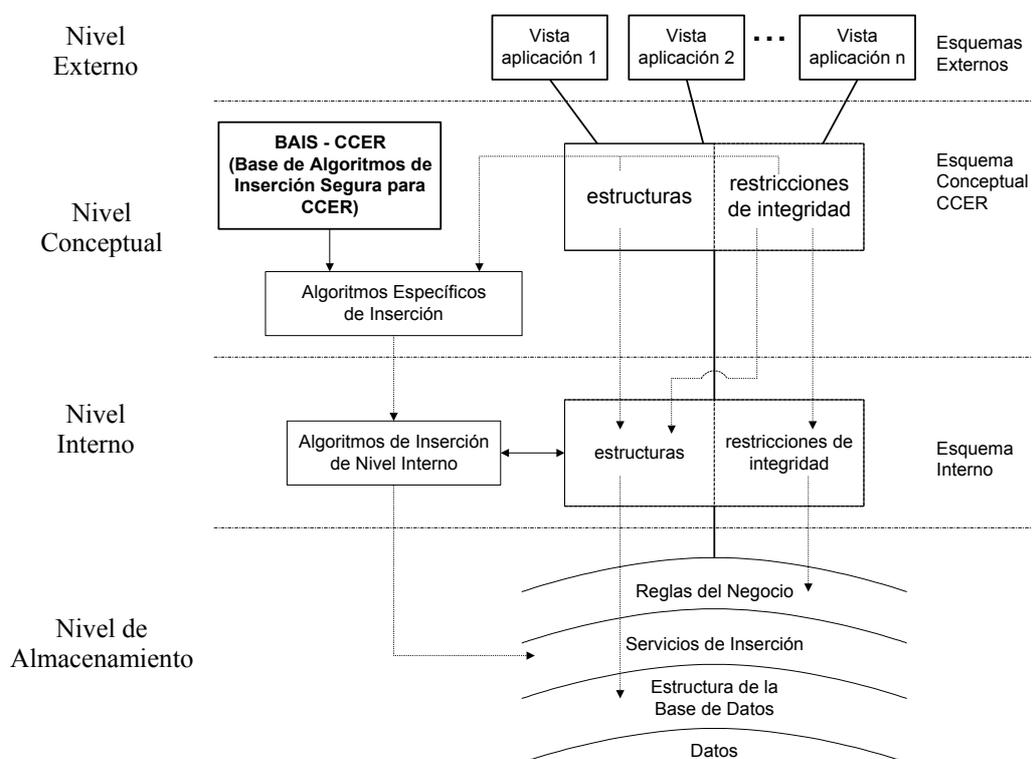


Figura 1: Arquitectura multinivel del Sistema de Bases de Datos para Inserción Segura

La arquitectura propuesta está compuesta de los siguientes niveles:

3.1.1 Nivel Externo

El nivel superior (*Nivel Externo*) contiene las vistas de las aplicaciones de usuarios que acceden a los datos de la base de datos, expresadas en el diagrama como Esquemas Externos.

3.1.2 Nivel Conceptual

En el *Nivel Conceptual*, se tiene el *Esquema Conceptual* en CCER de la base de datos, que contiene las estructuras y restricciones de integridad expresadas en forma independiente del Sistema de Gestión de Bases de Datos (SGBD) particular, mediante los formalismos gráfico y textual del Modelo CCER.

A partir del mismo modelo, fue generada la *Base de Algoritmos de Inserción Segura para CCER* (BAIS-CCER), un conjunto de algoritmos que permiten insertar datos sin violar las restricciones de integridad de un esquema en particular. De este conjunto de algoritmos, y a través de un mapeo de estructuras y restricciones del esquema conceptual, se obtienen los *Algoritmos Específicos de Inserción Segura*, que son los algoritmos para realizar la inserción en la base de datos específica que se está modelando.

3.1.3 Nivel Interno

Las estructuras definidas en el nivel conceptual son mapeadas a estructuras en el *Nivel Interno*, donde se establece la estructura de almacenamiento y recuperación de datos. Las restricciones del nivel conceptual se mapean a estructuras y restricciones de este nivel, comprendidas en el llamado *Esquema Interno*.

Los Algoritmos Específicos de Inserción Segura (nivel conceptual) se mapean a nivel interno, obteniendo los *Algoritmos de Inserción de Nivel Interno*, mapeo que depende de las estructuras y restricciones obtenidas en este nivel.

3.1.4 Nivel de Almacenamiento de Datos

El nivel de almacenamiento de datos se organiza en un conjunto de capas, siendo la más interna la correspondiente a los datos y la más externa a las Reglas de Mantenimiento de Consistencia. El dinamismo o variabilidad de la información almacenada explica este orden, variabilidad que aumenta desde las capas internas hacia las externas.

Las *Reglas de Mantenimiento de Consistencia* corresponden a las hasta aquí llamadas restricciones de integridad, pero almacenadas de acuerdo al SGBD particular. Se obtienen mediante mapeo desde las restricciones de integridad de nivel interno. Caso análogo lo constituye la *Estructura de la Base de Datos*, mapeada desde las estructuras del esquema interno.

Finalmente, los Algoritmos de Inserción de Nivel Interno son mapeados al nivel de almacenamiento, dando origen a los *Servicios de Inserción*, que son los procedimientos que llevarán a cabo la inserción de los datos. Estos Servicios ocupan una capa intermedia, bajo las Reglas de Mantenimiento de Consistencia y sobre la Estructura de la Base de Datos. Esto se explica por el hecho que cualquier modificación en tales reglas, que corresponden a la capa con mayor variabilidad dentro del nivel de almacenamiento, implicará una modificación de los Servicios de Inserción, no así de la Estructura de la Base de Datos.

4 Diseño de Algoritmos

La definición de restricciones de integridad incluye un conjunto de condiciones a ser cumplidas en todo momento por la base de datos. También incluye, explícita o, como en el caso de este estudio, implícitamente, un conjunto de transiciones legales entre los estados de la base de datos, legalidad definida por dicho cumplimiento.

La generación de algoritmos de inserción segura a partir de especificaciones formales de restricciones de integridad, implica determinar estas transiciones legales y expresarlas en un algoritmo. Este algoritmo no sólo debe contemplar todas las transiciones, sino expresarlas funcionalmente de modo de proveer una herramienta que, junto con asegurar la consistencia de la base de datos posterior a su utilización, reúna características de usabilidad y facilidad de mapeo a niveles posteriores del desarrollo de la base de datos. Soslayar estas características limitaría la utilización real de los algoritmos definidos, obteniendo instrumentos de chequeo con un índice de utilidad bajo.

El proceso de diseño de estos algoritmos implica estar permanentemente alerta para el bloqueo de las posibles fuentes de inconsistencias. En esta etapa, se toman decisiones de prioridades en cuanto a la ejecución de las condiciones de cumplimiento de la restricción. Un algoritmo puede considerar todas estas condiciones, pero este orden determinará, finalmente, si se cumplen a cabalidad.

El diseño de los algoritmos es consistente con la definición de las estructuras conceptuales a manipular: entidades e

interrelaciones. Se consideró como estructura base a la entidad. Así, tanto para inserciones en Tipo de Entidad como en Tipo de Interrelación, son entidades los elementos que se reciben como parámetros. En el primer caso, previo cumplimiento de las condiciones de la restricción, se insertan entidades; en el segundo, dichas condiciones son chequeadas tomando en cuenta la agregación de las entidades recibidas, agregación que conformará la interrelación a insertar. El nivel conceptual de los algoritmos, entonces, no sólo se relaciona con el origen de los algoritmos, sino en el tratamiento de los datos a insertar.

El criterio fundamental que rige el diseño de los algoritmos de la BAIS-CCER es privilegiar la consistencia de la base de datos por sobre todo otro aspecto. La interacción con el usuario se limitó a la elección de los datos a insertar, tomando cada algoritmo la responsabilidad de orientar dicha interacción, aumentando la complejidad de la tarea de diseño.

La característica de forzar el cumplimiento de las restricciones se traduce en un carácter inflexible de los algoritmos, obligando en muchos casos a contar con todos los datos necesarios para dicho cumplimiento. Privilegiando el estricto cumplimiento de las restricciones de integridad, se soslayaron aspectos de usabilidad. Esto puede dificultar la utilización inmediata de los algoritmos en aplicaciones comunes de bases de datos. Una de las próximas tareas a realizar es dotar a estos algoritmos de la flexibilidad requerida por el usuario para ejecutar procedimientos transaccionales que implementen en forma segura las restricciones de integridad.

5 Mecanismo de Mapeo BAIS-CCER y Estrategias de Ejecución

5.1 Mecanismo de Mapeo Restricción-Algoritmo

Para utilizar los algoritmos diseñados en casos concretos de diseño conceptual de bases de datos, se debe utilizar un mecanismo de mapeo que asocie instancias de restricciones de integridad a sus correspondientes algoritmos.

El mecanismo de mapeo de restricciones a algoritmos de inserción es un conjunto de reglas que permite:

- Identificar una restricción de integridad particular, en base a su especificación formal.
- Recuperar el algoritmo de inserción segura asociado a dicha restricción, desde la BAIS-CCER.
- Establecer la asociación entre las estructuras involucradas en la restricción y los argumentos del algoritmo.

El mecanismo definido considera métodos que permitan asociar:

- El formalismo textual del modelo CCER al lenguaje utilizado para la definición de los algoritmos.
- Las estructuras del esquema conceptual en CCER a los parámetros y variables de cada algoritmo.
- Las restricciones del esquema CCER a los operadores y condiciones de los algoritmos.

5.2 Estrategias de Ejecución

Para la ejecución de los algoritmos de la BAIS-CCER, se consideró especialmente el proceso de poblamiento de las bases de datos, terreno tradicionalmente fértil para la aparición de inconsistencias, debido a la extendida práctica de suspender temporalmente los mecanismos de chequeo durante este proceso. El tratamiento especial del poblamiento se justifica plenamente, pues es aquí donde las transacciones se limitan, salvo en casos excepcionales, al ingreso de datos, es decir, donde se ejecutarán exclusivamente los algoritmos de inserción. Las estrategias de ejecución de los algoritmos para el poblamiento permitirá obtener un estado consistente de la base de datos, a partir del cual las siguientes inserciones tendrán la responsabilidad de mantener dicho estado.

Se prioriza la ejecución de algoritmos de chequeo de dominio para atributos antes que cualquier otra ejecución. Esto permite comprobar de inmediato si los valores ingresados son reales candidatos a convertirse en atributos, otorgando argumentos validados a los algoritmos restantes. De esta forma, se construyen las entidades a partir de datos al menos sintáctica y semánticamente correctos, restando verificar el cumplimiento de las restricciones más complejas.

Al ejecutarse, los algoritmos de la BAIS-CCER, reciben parámetros ya ingresados por un procedimiento principal, donde se selecciona la estructura en donde se realizará la inserción y, de acuerdo a tal elección, se ingresan los valores de los atributos correspondientes, en el caso de los Tipos de Entidad, o se seleccionan las entidades que participarán en la interrelación a insertar, en el caso de los Tipos de Interrelación.

El poblamiento de la base de datos se centra en la inserción de entidades. Al ser una interrelación una agregación de entidades, el poblamiento de los Tipos de Interrelación se realiza seleccionando entidades ya insertas en los Tipos de Entidad involucrados. Tanto en el caso de Tipos de Entidad como Tipos de Interrelación, el poblamiento se realiza considerando restricciones de integridad básicas consideradas en el modelo CCER, que limitan la cantidad de instancias a insertar en dichas estructuras. Tratamientos especiales se adoptan para el poblamiento de Tipos de

Entidad que participen de una Generalización y para Tipos de Interrelación cuyas instancias estén sujetas a restricciones de dependencia o cardinalidad.

Teniendo la base de datos poblada y consistente, de acuerdo a la estrategia recién descrita, la ejecución de los algoritmos de inserción cuenta con más grados de libertad. En cuanto al orden de ejecución, se sugiere priorizar los algoritmos correspondientes a restricciones más complejas o que involucren a un número mayor de estructuras, precediendo a aquellos que implementen restricciones más simples.

6 Ejemplo de Aplicación

En esta sección, se mostrará la aplicación de esta propuesta para una restricción de integridad básica en el diseño conceptual de bases de datos: la cardinalidad de Tipo de Entidad con respecto a Tipo de Interrelación. Se presenta una descripción de los requerimientos que la restricción impone al algoritmo de inserción correspondiente, el diagrama de flujo de dicho algoritmo y su expresión en pseudo-lenguaje, mostrando además un ejemplo concreto de la restricción y de la ejecución del algoritmo diseñado.

6.1 Descripción del algoritmo

6.1.1 Requerimientos funcionales para el algoritmo de inserción

Para la asociación de dos o más entidades en una interrelación, se debe verificar el cumplimiento de las cardinalidades mínima y máxima de sus respectivos Tipos de Entidad con respecto al Tipo de Interrelación donde se desea insertar dicha asociación.

Para asegurar el cumplimiento de esta restricción, el algoritmo de inserción que la implemente debe considerar los siguientes pasos:

- Todas las entidades a agrupar en el Tipo de Interrelación R_X deben ya estar insertas en sus respectivos Tipos de Entidad (*chequeo de integridad referencial*). Si esto no ocurre, la operación debe rechazarse.
- Se realiza un *chequeo de unicidad*, para asegurar que la interrelación (ya validada) no ha sido previamente insertada en R_X . Si esto ya ha ocurrido, se rechaza la transacción.
- Con la inserción deseada, cada una de las entidades que se quiere agregar en interrelación debe cumplir con las cardinalidades máximas de los Tipos de Entidad con respecto a R_X , es decir, previo a la inserción, ninguna entidad debe haber llegado a su límite máximo de participación en R_X . Basta que se dé uno solo de estos casos para rechazar la inserción.
- Superados satisfactoriamente los chequeos anteriores, se forma la interrelación, a partir de las entidades y valores de atributos propios de R_X (en CCER, *subespectro* de R_X) y se inserta.
- Cada una de las entidades agregadas en interrelación debe alcanzar, con la inserción realizada, una cantidad de participaciones a lo menos igual a la cardinalidad mínima del Tipo de Entidad correspondiente con respecto a R_X . Si es así, el algoritmo concluye, si no, se obliga al usuario a seleccionar las entidades que, yuxtapuestas con la entidad analizada, forme una interrelación que será insertada en R_X .

6.1.2 Entradas

- R_X : Tipo de Interrelación donde se desea realizar la inserción.
- $e_1, \dots, e_{\text{grado}(R_X)}$: entidades que participan en la interrelación a insertar. El grado de R_X ($\text{grado}(R_X)$, operador definido en [10]) es la cantidad de Tipos de Entidad que se relacionan a través de R_X . Estos Tipos de Entidad conforman el *espectro* de R_X .
- $f_1, \dots, f_{\text{grado}(R_X)}$: entidades que conforman interrelaciones adicionales a insertar, para cumplir con cardinalidades mínimas de sus respectivos Tipos de Entidad con respecto a R_X .

6.1.3 Salidas

- R_X con interrelación(es) insertada(s)
- Señales de error por intento de incumplimiento de cardinalidad máxima, integridad referencial o unicidad

6.1.4 Diagrama de Flujo de Algoritmo

En la siguiente figura, se pueden apreciar los sucesivos chequeos que permiten comprobar el cumplimiento de las condiciones de la inserción para esta restricción. La primera estructura de selección corresponde al chequeo de integridad referencial, mientras que la segunda comprueba la inexistencia previa de la interrelación a insertar. Las estructuras restantes se centran específicamente en la restricción de cardinalidad.

Eventualmente, para que la inserción de una interrelación produzca un estado consistente en la base de datos, se requerirá la inserción de interrelaciones adicionales para alcanzar cardinalidades mínimas. Dichas interrelaciones deberán cumplir con las mismas condiciones que la inicial, por lo que se incluyó para este caso un llamado recursivo del algoritmo diseñado, como se muestra en la figura 2:

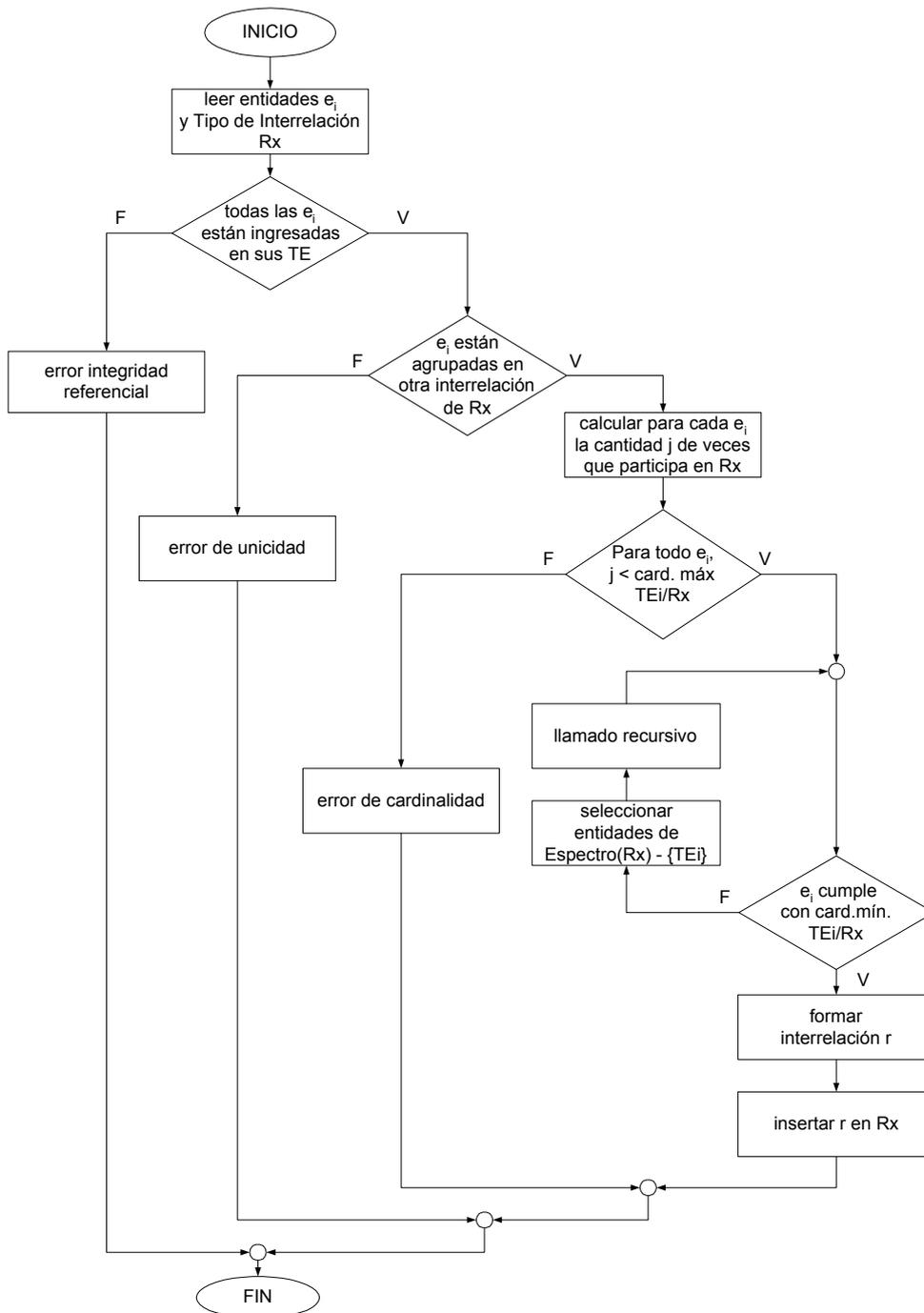


Figura 2: Algoritmo de Inserción Segura para Cardinalidad de Tipo de Entidad con respecto a Tipo de Interrelación

6.1.5 Algoritmo en Seudo-Lenguaje

Se muestra a continuación el algoritmo de la figura 2 expresado en seudo-lenguaje (Nota: considérese **R** como el conjunto de todos los Tipos de Interrelación incluidos en un Esquema Conceptual particular).

```

function insertion_cardterti (e1, ... , egrado(RX):entidad ; RX: R) : boolean
{
  tag ← true;
  gx ← grado (RX);
  ref_int ← ref_integrity (e1, ... , egx, Espectro(RX));
  if (not ref_int) then
    {
      error ('interrelación con al menos una entidad no insertada');
      return false;
    }
  else
    if (inserted (e1, ... , egx, RX)) then
      {
        error ('interrelación ya presente en RX');
        return false;
      }
    else
      {
        j ← 1;
        while ((howmany(ej, RX, TEj) < maxCard(TEj, RX)) and (j ≤ gx)) do
          j ← j+1;
        if ((j > gx) or (maxCard(TEj, RX) = n)) then
          {
            k ← 1;
            while (k ≤ gx) do
              {
                if ((howmany (ek, RX, TEk) + 1) < minCard (TEk, RX)) then
                  {
                    for i ← 1 to gx do
                      if (i ≠ k) then
                        fi ← select (entidad de TEi ∈ Espectro (RX) - {TEk});
                        tag ← insertion_cardterti (f1, ..., fk-1, ek, fk+1, ..., fgx, RX);
                  }
                if (tag) then
                  k ← k+1;
              }
            r ← formar_inter (e1, ..., egx, RX);
            insert (r, RX);
            return true;
          }
        else
          {
            error ('al menos una de las entidades alcanzó cardinalidad máxima c/r a RX');
            return false;
          }
      }
    }
}

```

Es posible distinguir algunos operadores propios del lenguaje de diseño de la BAIS-CCER, por ejemplo:

- minCard (TE_i, R_Y), maxCard (TE_i, R_Y): funciones enteras ($\cup \{n\}$) que retornan la cardinalidad mínima y máxima del Tipo de Entidad TE_i con respecto al Tipo de Interrelación R_Y.
- ref_integrity (e₁, ..., e_n, TE₁, ..., TE_n): función buleana que chequea la existencia actual de las entidades e_i en sus respectivos Tipos de Entidad TE_i, i=1..n

- $inserted(e_1, \dots, e_n, R_Y)$: función booleana que verifica si existe alguna interrelación, actualmente inserta en el Tipo de Interrelación R_Y , que agrupe las entidades e_1, \dots, e_n .
- $howmany(e_j, R_Y, TE_j)$: función entera que retorna el número de interrelaciones en el Tipo de Interrelación R_Y que incluyen a la entidad e_j perteneciente al Tipo de Entidad TE_j .
- $formar_inter(e_1, \dots, e_n, R_X)$: función que retorna una interrelación formada a partir de los identificadores de las entidades e_i y valores que el usuario ingresa para atributos propios del Tipo de Interrelación R_X , previo chequeo de dominio.

6.2 Ejemplo de Uso

Se tienen, entre otros, los siguientes requerimientos para el diseño de una base de datos que permitirá manejar información sobre concursos para proyectos de investigación:

- Dos veces al año se llama a concurso de proyectos de investigación.
- Existe la definición de áreas prioritarias por concurso. Cada concurso considera una cantidad de áreas prioritarias no inferior a 2 ni superior a 5.

En la figura 3, se muestran estos requerimientos expresados en el formalismo gráfico de CCER (en este caso, muy similar a la expresión en MER extendido):

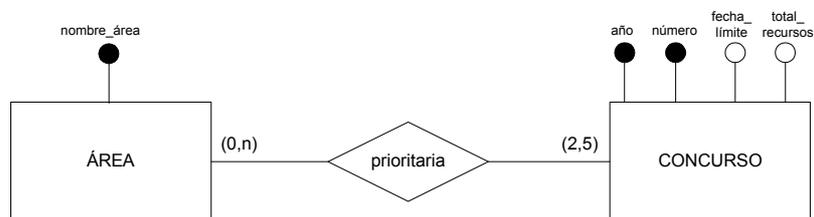


Figura 3: Ejemplo de Cardinalidad de Tipo de Entidad con respecto a Tipo de Interrelación, en CCER Gráfico.

La siguiente es la especificación de las estructuras y restricciones de este caso, expresada en el formalismo textual de CCER:

[dominios]

- IN = {x / x es número natural}
 FECHA1 = {x / x es una fecha válida y x es posterior a 31-dic-1999}
 AREA_INV = {x / x es el nombre de un área de investigación}
 FECHA2 = {x / x ≥ 2000}
 UNODOS = {x / (x = 1) ∨ (x = 2)}

[atributos simples]

- nombre_área ⊆ AREA_INV
 año ⊆ FECHA2
 número ⊆ UNODOS
 fecha_límite ⊆ FECHA1
 total_recursos ⊆ IN

[tipos de entidad]

- ÁREA ⊆ nombre_área, I_ÁREA = nombre_área
 CONCURSO ⊆ año x número x fecha_límite x total_recursos, I_CONCURSO = año x número

[tipos de interrelación]

- prioritaria ⊆ (CONCURSO x ÁREA)

[cardinalidades binarias]

- card (CONCURSO, prioritaria) = (2,5)
 card (ÁREA, prioritaria) = (0,n)

A partir de esta especificación textual, el algoritmo diseñado recoge las instancias para sus parámetros y los valores retornados por sus operadores.

Se requiere realizar la siguiente operación: establecer, como área prioritaria para el primer concurso del 2002, al área de “*Web Semántica*”

Suponiendo que el concurso *2002-1* y el área *Web Semántica* ya fueron insertados en CONCURSO y ÁREA, respectivamente, y que dicha área no está ya calificada como prioritaria para el *2002-1*, los chequeos de integridad referencial y unicidad se cumplen satisfactoriamente.

Se procede a contar la cantidad de áreas prioritarias ya asignadas al concurso *2002-1*. Si dicha cantidad es 5 (cardinalidad máxima de CONCURSO con respecto a *prioritaria*), ya se llegó al límite superior de áreas prioritarias para dicho concurso, por lo que la asignación deseada se rechazaría. Supóngase que eso no ocurre. Se realiza el similar proceso para conocer la cantidad de concursos en que *Web Semántica* es prioritaria, pero en este caso, no existe límite superior de asignaciones (cardinalidad máxima n), por lo que esto no restringe la asignación.

Se consulta ahora, para cada entidad, si con la asignación a ingresar se cumplirá con la cardinalidad mínima con respecto a *prioritaria*. Si no es así, se solicitará la selección de entidades para completar una nueva participación de la entidad en la interrelación, hasta cumplir con la cantidad mínima exigida. Supongamos que no se han asignado áreas prioritarias a *2002-1*. Con la asignación de *Web Semántica*, aún no se llegaría a las 2 áreas que, como mínimo, se requiere para el concurso, por lo que se selecciona de las áreas ingresadas a *Data Mining*, otra área prioritaria para el *2002-1*.

Se realiza similar proceso, mediante llamado recursivo, para esta nueva instancia: se verifica que *Data Mining*, con su asignación, alcance su correspondiente cardinalidad mínima con respecto a *prioritaria*. Como dicha cardinalidad mínima es cero, no habrá necesidad de asignarla a otro concurso. Por lo que se ingresa la asignación de *Data Mining* como área prioritaria para el concurso *2002-1*, retornando al anterior llamado de la lógica recursiva.

Se realiza ahora el mismo chequeo para *Web Semántica*, que tampoco requiere nuevas asignaciones, por lo que, finalmente, se ingresa la asignación de esta área como prioritaria para el concurso *2002-1*.

7 Conclusiones

La propuesta presentada ha logrado fortalecer un modelo conceptual de datos con un conjunto de algoritmos que permitirán realizar inserciones seguras, respetando las restricciones de integridad consideradas en los correspondientes esquemas de datos. Mediante un mecanismo de mapeo restricción-algoritmo, se podrá utilizar estos algoritmos en esquemas particulares, identificando las restricciones incluidas y asociándolas con sus correspondientes algoritmos de inserción.

Para abordar el problema global de inserción de la base de datos, se generaron estrategias de ejecución de los algoritmos para su poblamiento y para inserciones posteriores, basadas en las relaciones existentes entre las restricciones consideradas en este estudio.

La generación de algoritmos de inserción directamente desde las especificaciones de restricciones de integridad incluidas en el esquema conceptual de la base de datos, implica una concepción de la base de datos orientada a las estructuras de nivel conceptual. Se habla de inserciones en Tipos de Entidad y en Tipos de Interrelación, rigiéndose dichas inserciones por las múltiples formas de inclusión de dichas estructuras en esta fase del desarrollo.

Los aportes de esta propuesta se pueden resumir en los siguientes puntos:

- El diseño de algoritmos de inserción orientados al cumplimiento de las restricciones de integridad incluidas en el nivel conceptual, permitirá contar con una base de datos consistente desde su poblamiento. Para las inserciones de datos posteriores, los algoritmos mantendrán dicho estado consistente o, si la base de datos ya sufre de inconsistencias producto de otro tipo de transacciones, no introducirá otras nuevas.
- La generación de procedimientos de manipulación de datos directamente del esquema de la base de datos permitirá que eventuales actualizaciones de tales procedimientos, producto de modificaciones y depuraciones en los esquemas de datos, sean de mucho menor costo para los desarrolladores que si aquellos se hubiesen construido en las últimas etapas.
- El diseño de algoritmos de inserción a nivel conceptual permite su utilización en múltiples implementaciones particulares de la base de datos modelada, independientemente de consideraciones de diseño en niveles posteriores del desarrollo. De este modo, se evita la duplicación de esfuerzos en la consideración de las restricciones, por ejemplo, para procesos de migración o actualización de plataforma de implementación. Sólo las decisiones que se tomen en el nivel conceptual afectarán los algoritmos obtenidos.

- El modelo CCER, mediante la especificación formal de estructuras y de un conjunto importante de restricciones de integridad, otorga los recursos necesarios para generar completamente los métodos de inserción. El proyecto constituye un aporte al desarrollo de bases de datos utilizando CCER, al dotarlo de procedimientos algorítmicos que consideran las restricciones de integridad expresables en este modelo.
- Las directrices para la ejecución de los procedimientos algorítmicos, que orientarán la ejecución de éstos, permitirá minimizar conflictos que pudieran surgir en el cumplimiento de las múltiples restricciones presentes en el esquema conceptual de la base de datos.

Algunos trabajos posibles de desarrollar a partir de esta propuesta son los siguientes:

- Desarrollo de mecanismos de mapeo a nivel lógico de los algoritmos de la BAIS-CCER, considerando su utilización en base a los componentes del Modelo Relacional y del Diseño Orientado a Objetos.
- Generación directa de procedimientos algorítmicos, a partir de la especificación de restricciones de integridad, para realizar otras operaciones la base de datos, como actualización y eliminación de datos.
- Incorporación, a los procedimientos generados, de detecciones y correcciones de esquemas CCER erróneos, es decir, con inconsistencias entre las restricciones incluidas.
- Transformación de la Base de Algoritmos en una Base de Conocimiento, implementando las directrices de ejecución establecidas en este proyecto. Es decir, la obtención de un almacén de procedimientos y de reglas de conocimiento para efectuar automáticamente relaciones y/o combinaciones de algoritmos para casos particulares, de tal modo que se permitan generar servicios específicos de inserción para una base de datos cualquiera, a partir de su esquema de datos.
- Diseño e implementación del mecanismo de mapeo restricción-algoritmo, recogiendo conceptos de agentes inteligentes y haciendo uso de la futura Base de Conocimiento BAIS-CCER.

Referencias

- [1] Batini, C., Ceri, S. y Navathe, S. *Diseño Conceptual de Bases de Datos: Un enfoque de entidades-interrelaciones*. Addison-Wesley/Díaz de Santos, 1994.
- [2] Balaban M. y Shoval, P. *MEER – An EER Model Enhanced with Structure Methods*. Information Systems Program, Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev, Israel, 1999.
- [3] Cammarata, S., Ramachandra, P., Shane, D. Extending a Relational Database with Deferred Referential Integrity Checking and Intelligent Joins. *ACM SIGMOD Record, Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data* (junio, 1989), pp. 88-97.
- [4] Celle, A., Bertossi, L. Consistent Data Retrieval. *Information Systems*, Elsevier Science Ltd., Vol.19, No.4, (1994), pp.33-54.
- [5] Ceri, S., Fraternali, P., Paraboschi, S., Tanca, L. Automatic Generation of Production Rules for Integrity Maintenance. *ACM Transactions on Database Systems*, Vol.19, No.3, (septiembre, 1994), pp.367-422.
- [6] Chen, P.P. The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Software Engineering*, Vol. 1, No.1, (marzo, 1976), pp.9-36.
- [7] Codd, E. F. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, Vol. 13, No. 6, (junio, 1970), pp. 377-387.
- [8] Korth, H., Silbershatz, A. *Fundamentos de Bases de Datos*. McGraw-Hill, 1993.
- [9] Sheard T., Stemple, D. (1989). Automatic Verification of Database Transaction Safety. *ACM Transactions on Database Systems*, Vol.14 No.3, (septiembre, 1989), pp.322-368.
- [10] Varas, M. *CCER: Constraint Centered Entity Relationship Model*. Monografía. Grupo AIMCON, Depto. de Ingeniería Informática y Ciencias de la Computación, Facultad de Ingeniería, Universidad de Concepción, Chile. <http://www.inf.udec.cl/~aimcon/monografia> (enero,2000).