

Abordajes alternativos para el rediseño de modelos de bancos de datos usando algoritmos genéticos y propagación de restricciones en grafo

Vinícius Medina Kern

Cristiano Dácio

Henrique José de Souza Coutinho

UNIVALI, UNITEC (Grupo de Pesquisa em Tecnologia e Sistemas)

Rodovia SC 407, Km 4; 88122-00; São José-SC, Brasil

Teléfono +55 (48) 281-1500, Fax +55 (48) 281-1506

kern@eps.ufsc.br, cidacio@sj.univali.br, hcoutinho@sj.univali.br

INDEXADORES

Algoritmos genéticos, diseño gráfico, proyecto de banco de datos, grafos, propagación de restricciones, modelo Entidad-Relacionamiento, IDEF1X

RESUMEN

El proyecto conceptual de banco de datos trata de capturar la estructura de la información en un ambiente. Un modelo de banco de datos es producto del proyecto conceptual y es usualmente representado en forma gráfica. Una característica importante de un modelo es su legibilidad, es decir, la calidad del arreglo de la disposición de símbolos gráficos de modo que la lectura por proyectistas y expertos en el ambiente modelado sea facilitada. Este artículo describe una investigación en realización cuyo objetivo es agregar a una herramienta gráfica de proyecto de banco de datos una función de rediseño. Dos abordajes son aventados: La formulación del problema como propagación de restricciones de diseño de modelos representados como grafos dirigidos y no cíclicos, y la formulación del problema como la aplicación de criterios de calidad sobre diseños alternativos y la selección del mejor diseño usando la técnica de algoritmos genéticos. El mecanismo de solución según los dos abordajes es discutido.

INTRODUCCIÓN

El proyecto conceptual es una etapa fundamental y crítica del proyecto de banco de datos. En ella se trata de capturar la estructura de la información en un ambiente de negocios. Un modelo de banco de datos es producto del proyecto conceptual y es usualmente representado en forma gráfica, según el Modelo Entidad-Relacionamiento (ER) [1] o uno de sus dialectos.

El modelo de un banco de datos representa las entidades (tipos de objetos), sus atributos o características, y los relacionamientos o asociaciones entre las entidades. Esos elementos expresan las reglas del negocio de nivel más alto, pues establecen formas de representación y restricciones sobre la **estructura** de la información.

El **diseño grafico** de modelo de banco de datos es el arreglo de la disposición de los símbolos gráficos del modelo en el plano. El diseño gráfico de un modelo de banco de datos presenta buena **legibilidad** si la disposición de los símbolos facilita la lectura por los proyectistas y expertos en el dominio modelado. Por ejemplo, si hay una regla en un ambiente universitario que dice que “Un departamento ofrece cero, uno, o varios cursos; Un curso es ofrecido por exactamente uno departamento”, entonces un diseño como el de la figura 1 facilita la lectura.

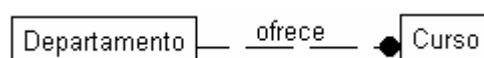


Figura 1 - Un modelo universitario simple

El problema se hace complejo cuando hay muchas entidades y relacionamientos. La figura 2 muestra dos diseños para un mismo modelo de banco de datos (todavía relativamente pequeño). El

primer (a) fue dibujado automáticamente por una herramienta comercial. El segundo (b) fue dibujado por un proyectista según reglas de buena legibilidad, presentadas en la próxima sección. Por razón desconocida, las funciones de rediseño eventualmente disponibles en herramientas comerciales no aplican reglas aceptas de buen diseño.

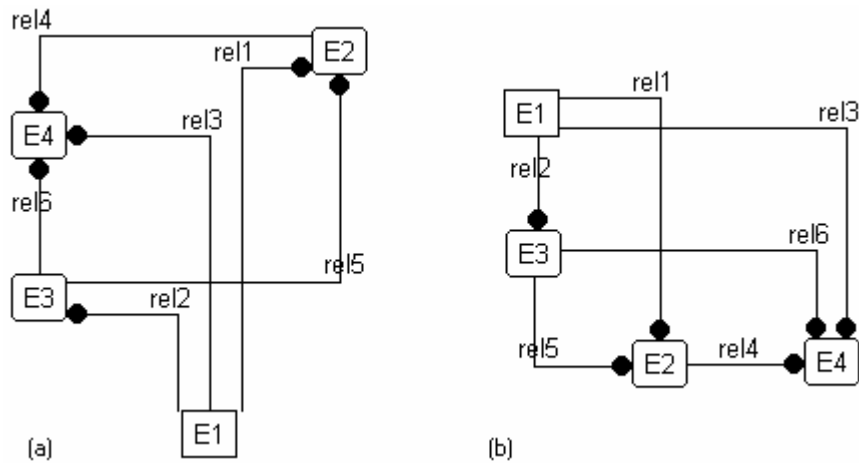


Figura 2 - Diseños hechos (a) automáticamente por una herramienta comercial y (b) por un proyectista de acuerdo con reglas de buena legibilidad

Problema y contribución

La principal contribución de este artículo es la presentación de la formulación del problema de rediseño de modelos de banco de datos y la propuesta de solución según dos abordajes alternativos: La propagación de restricciones de diseño de modelos representados como grafos dirigidos y no cíclicos, y la aplicación de criterios de calidad sobre diseños alternativos y la selección del mejor diseño usando la técnica de algoritmos genéticos. Esa formulación es resultado de un proyecto de investigación de iniciación científica.

La función de rediseño es una de las necesidades [2] de extensión de la herramienta IDEFClasses [3] [4], cuya interface es ilustrada en la figura 3. IDEFClasses apoya el proyecto de banco de datos según el lenguaje IDEF1X₉₇ o IDEFObject [5] para bancos de datos orientados a objeto.

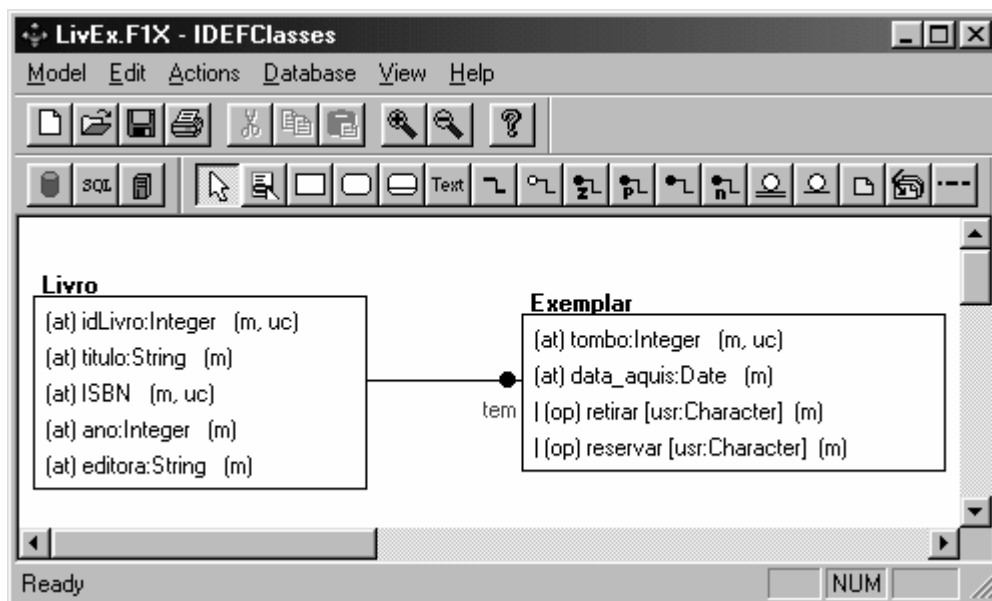


Figura 3 - La herramienta gráfica de proyecto de banco de datos IDEClasses [4]

La sección a seguir discute la cuestión de la legibilidad de modelos de banco de datos. Después, dos abordajes alternativos para el rediseño de modelos de banco de datos son presentados: la formulación de propagación de restricciones en grafos y la formulación de algoritmos genéticos. La conclusión, enseguida, resume el artículo y discute los próximos pasos.

LEGIBILIDAD DE MODELOS DE BANCO DE DATOS

Legibilidad es la característica de un buen diseño de modelo de banco de datos. Proyectistas, analistas, programadores y expertos en el dominio del negocio modelado necesitan leer, comprender, criticar, investigar, revisar y proponer alteraciones para un modelo en construcción.

Los proyectistas identifican y diseñan las estructuras de información. Los analistas y programadores implementan soluciones en computadora para las actividades de los usuarios del sistema de banco de datos. Los expertos en el dominio del negocio son responsables por revisar y validar el modelo. El proyecto avanza iterativamente. El problema de rediseño surge y se agrava cuando hay centenas o millares de entidades en el modelo. La necesidad de producir un nuevo diseño de un modelo de banco de datos puede ser caracterizada en dos situaciones:

- ? Alteración de diseño – los cambios en el modelo crean situaciones donde entidades están superpuestas y relacionamientos resultan ilegibles. El proyectista debe, entonces, crear un nuevo arreglo para la disposición de los elementos gráficos de modo que permita y facilite la lectura del modelo.
- ? Ingeniería reversa de modelo de banco de datos – un banco de datos existe pero no está disponible como modelo gráfico. El proyectista debe, en este caso, crear un arreglo para la disposición de los elementos gráficos de modo que permita y facilite la lectura del modelo.

La opción por IDEF1X

Chen [1] formuló el modelo ER, el abordaje más exitoso para el proyecto conceptual de banco de datos. IDEF1X [6] es un dialecto del modelo ER, basado también en el modelo relacional de datos [7], que incluye un lenguaje no-ambiguo con reglas para el arreglo de elementos gráficos.

Los elementos gráficos en IDEF1X son **cajas** (rectángulos), representando entidades, y **líneas** poligonales horizontales y verticales, representando relacionamientos. Atributos, cuando representados, tienen sus nombres inscritos en las respectivas entidades. La sintaxis y semántica de IDEF1X son presentadas informalmente y a través de una formalización usando lógica de primera orden [6].

El relacionamiento típico en IDEF1X es el uno-para-varios, también conocido como ancestral-descendiente (*parent-child*). La entidad ancestral está del lado “uno” y la descendiente del lado “varios”. En el ejemplo de la figura 1, Departamento es la entidad ancestral y Curso la descendiente. Relacionamientos varios-para-varios necesitan típicamente ser descompuestos en dos relacionamientos uno-para-varios de las entidades participantes con una nueva entidad asociativa.

La regla para diseño en IDEF1X es: la entidad descendiente debe estar más abajo y/o más a la derecha de la entidad ancestral. De esta forma, cada relacionamiento ancestral-descendiente puede ser leído en la manera occidental: de la izquierda para la derecha, o desde arriba para abajo. El relacionamiento de la figura 1 respecta esta regla, y también los relacionamientos de la figura 2 (b). En la figura 2 (a), sin embargo, la regla no es respetada.

Es más fácil analizar un modelo con buena legibilidad: Todos los relacionamientos uno-para-varios pueden ser leídos en la manera occidental en la forma: <Artículo indefinido> <entidad ancestral> <relacionamiento> <multiplicidad> <entidad descendiente>. Eso corresponde a la estructura sintáctica <Artículo indefinido> <sustantivo (sujeto)> <verbo transitivo o frase verbal> <multiplicidad> <sustantivo (objeto)>. El ejemplo ya citado, de la figura 1, puede ser leído como “Un departamento ofrece cero, uno, o varios cursos”.

Otra manera de garantizar que cada relacionamiento acoge la regla de buena legibilidad es verificar que el punto negro (junto al lado “varios”) está más abajo y/o a la derecha que el origen (junto al lado “uno”) en la entidad ancestral. Eso y más reglas de sentido común ayudan a llegar a un modelo gráfico con buena legibilidad. Las reglas de sentido común tratan de mantener la facilidad de lectura: no sobreponer entidades, evitar que entidades relacionadas se ubiquen demasiado cercanas o

lejanas, evitar que líneas de relacionamiento se superpongan con otras líneas o entidades. En resumen, un diseño legible tiene las cuatro características siguientes:

1. Cada entidad ancestral debe estar más arriba y/o a la izquierda que sus descendientes,
2. ninguna entidad debe estar superpuesta a otra,
3. entidades asociadas no deben estar muy cercanas ni muy lejanas, y
4. líneas de relacionamientos no deben superponer a otros relacionamientos ni a entidades.

Dar legibilidad a un modelo problemático equivale a alterar el diseño de modo que asegure la obediencia a las reglas recién presentadas. Para tanto, las próximas secciones proponen dos abordajes para solucionar el problema, basadas en propagación de restricciones en grafos y en algoritmos genéticos.

REDISEÑO POR PROPAGACIÓN DE RESTRICCIONES EN GRAFOS

Modelos de banco de datos han sido estudiados de forma análoga a grafos dirigidos no-cíclicos [8]. Las entidades pueden ser consideradas los nodos del grafo, y los relacionamientos representan la dirección y el sentido de las ligaciones, según la orientación ancestral-descendiente.

Un mecanismo de rediseño de modelos gráficos de banco de datos puede usar las cuatro restricciones de legibilidad y propagarlas sobre el grafo. El punto de partida puede ser la localización de todas las entidades en un mismo punto y entonces, la aplicación un distanciamiento o *offset* entre cada dos entidades ancestral y descendiente, conforme ilustra la figura 4, relativamente a la primera restricción de legibilidad. Enseguida, se puede verificar las demás tres restricciones y ajustar la disposición espacial de las entidades.

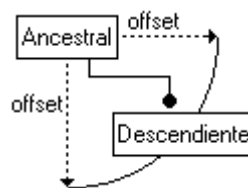


Figura 4 – Posición de una entidad descendiente con relación a una entidad ancestral

Hay que escoger un valor para la distancia mínima (*offset*) que sea suficiente para que cada relacionamiento pueda ser leído. Es importante observar, también, que este abordaje es sub-restricto. Así, será necesario usar alguna heurística o regla intuitiva para escoger si una entidad descendiente será posicionada más para la derecha, o más para abajo o, aún, más abajo y más a la derecha que su ancestral. La figura 5 muestra tres diseños legibles distintos para el mismo modelo de banco de datos.

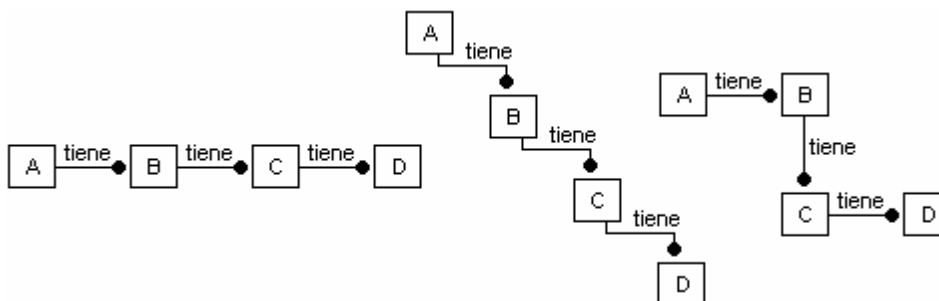


Figura 5 – Varios diseños legibles alternativos para el mismo modelo de banco de datos

Es muy posible que un mecanismo automático basado en este principio necesite ser realizado de modo iterativo. Después de garantizar el cumplimiento de las cuatro restricciones de legibilidad, se puede pensar en criterios adicionales como la minimización del área total ocupado por el diseño.

Considerando que los diseños frecuentemente son impresos y pegados en una pared, la alternativa de la derecha en la figura 5 parece más interesante.

A pesar de la falta de restricciones y la consecuente necesidad de creación artificial de restricciones para llegar a un diseño, este abordaje puede ser considerado bastante tradicional, se comparado al abordaje de algoritmos genéticos, discutido en la próxima sección.

REDISEÑO POR ALGORITMOS GENÉTICOS

La técnica de algoritmos genéticos tiene en Goldberg [9] una de sus principales referencias. Formular un problema según la técnica de algoritmos genéticos equivale a crear soluciones aleatorias (satisfactorias o no) para el problema en la forma de “individuos” y, entonces, promover la evolución de las soluciones en sucesivas generaciones, emulando la selección natural darwiniana.

Problemas de disposición espacial han sido abordados usando algoritmos genéticos por Wilson [10] y Coutinho *et alii* [11]. Mangano [12] sugiere el uso de algoritmos genéticos para problemas de diseño de grafos, como ya vimos que puede ser formulado el rediseño de modelos de banco de datos.

Formulación del problema por algoritmos genéticos

Siempre que se aplica la técnica, soluciones iniciales son escritas como **individuos** formados por **cromosomas**. La figura 6 ilustra la formulación del problema de diseño de modelos de banco de datos. Cada individuo es un diseño. Cada entidad en cada individuo corresponde a un **cromosoma**, para el cual se anotan dos valores representan su posición en el plano. El individuo puede ser representado fácilmente como una cadena de caracteres.

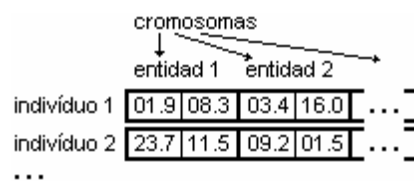


Figura 6 – Formulación de diseños de un modelo de banco de datos como individuos para abordaje al rediseño usando algoritmos genéticos

Un conjunto de individuos forma una **población**. Las poblaciones son compuestas por diseños con variados niveles de legibilidad. Cada nueva **generación** de individuos es producida a través de cruces y mutaciones en la población seleccionada, y los individuos más aptos son escogidos para tener descendencia. La evolución sigue hasta que se llegue a una solución (individuo) satisfactoria, es decir, hasta que las nuevas generaciones no puedan generar individuos más aptos que los ya existentes.

La aptitud de un individuo es investigada por una **función de aptitud** (*fitness*) que penaliza individuos que no respetan a las restricciones de legibilidad, es decir, diseños en los cuales las entidades y relacionamientos están muy cercanos o sobrepuestos, y donde los relacionamientos uno-para-varios no pueden ser leídos a la manera occidental.

Cada violación de restricción de legibilidad recibe un peso negativo. Cada individuo recibe un índice de aptitud que es función de las violaciones a las reglas de legibilidad. Los mejores individuos son seleccionados a cada generación y se producen nuevos individuos a partir de los existentes, hasta que se llegue a un nivel estable de los valores mínimos de la función de aptitud, cuando se escoge el mejor diseño.

CONCLUSIÓN

En este artículo, presentamos el problema de rediseño de modelos gráficos de banco de datos y la importancia de la legibilidad de los modelos. Criterios de buena legibilidad fueran expuestos, y dos abordajes posibles para el rediseño fueron aventados, basados en propagación de restricciones en grafos y en algoritmos genéticos.

En la etapa actual de investigación se ha implementado la solución por algoritmos genéticos, sin embargo no se ha logrado hasta el momento un ajuste de pesos de penalización de individuos que lleve a solución siempre satisfactorias. En las próximas etapas se proyecta seguir experimentando con criterios de selección de diseños y respectivos pesos, y también pasar a implementar la solución tradicional basada en propagación de restricciones en grafos, para enseguida compararlas y elegir la mejor o adoptar a ambas, si ninguna resultar siempre superior.

Esta investigación brinda posibilidades de incluir otras técnicas en el proceso de construcción o selección de diseños con buena legibilidad, especialmente en el caso del rediseño por propagación de restricciones en grafo. Algunas preguntas que podrían ser respondidas quizás a través de técnicas de inteligencia artificial son: ¿Como haz un proyectista para decidir si una entidad descendiente va a ser posicionada más abajo o más a la derecha (o más abajo y más a la derecha) de su ancestral? ¿Como hacer para evitar que algún par ancestral-descendiente quede muy distante mientras se respeta las demás reglas de legibilidad? Considerando tratarse de una técnica sub-restrita, algún criterio debe ser usado para llegar a un diseño definido. Quizás este tipo de criterio guarde el secreto de la buena legibilidad.

AGRADECIMIENTOS

Esta investigación recibió apoyo, en esa etapa inicial, del gobierno del estado de Santa Catarina, Brasil, a través de los recursos de la Constitución Estadual, artículo 170. El señor Dácio es estudiante de Ingeniería de Computación. Tuvo como orientador a Dr. Coutinho, con la colaboración del Dr. Kern.

REFERENCIAS

- [1] P.P. Chen, The Entity-Relationship model - toward a unified view of data. **ACM Transactions on Database Systems** 1 (1), p. 9-36, 1976.
- [2] V. M. Kern y M. U. Bragaglia. Oportunidades de desenvolvimento e pesquisa sobre projeto de bancos de dados usando IDEF1X. **Alcance** ano VI, n. 3, p. 50-57. Itajaí: Editora da UNIVALI, 1999.
- [3] **IDEFClasses**: An IDEFobject modeling interface. Sitio en la Internet dedicado a la herramienta de proyecto de banco de datos IDEFClasses. Disponible en <<http://www.eps.ufsc.br/~kern/idefclasses/idefclasses.html>>. Acceso en 27 sep. 2001.
- [4] M. U. Bragaglia. **Software para modelagem de bancos de dados objeto-orientados usando IDEFobject**. Trabalho de Conclusão do Curso de Ciência da Computação. São José: UNIVALI, jun. 1999.
- [5] IEEE (Institute of Electrical and Electronic Engineers) IDEF1X Standards Working Group. **Standard for conceptual modeling language syntax and semantics for IDEF1X97 (IDEFobject)**. IEEE 1320.2 Standards Committee, document P1320.2, release draft 0.91, 314 p., May 1, 1998.
- [6] NIST (National Institute of Standards and Technology). **Federal Information Processing Standards Publication 184: Integration Definition for Information Modeling (IDEF1X)**, Gaithersburg, MD, December 1993.
- [7] E.F. Codd. A relational model of data for large shared data banks. **Communications of the ACM** 13 (6), p. 377-387, 1970.
- [8] M. F. van Bommel y G. E. Weddell. Reasoning about equations and functional dependencies on complex objects. **IEEE Transactions on Knowledge and Data Engineering** 6 (3), June 1994.
- [9] D. E. Goldberg. **Genetic algorithms in search, optimization, and machine learning**. Addison-Wesley, 1989.
- [10] S. Wilson. How to grow a starship pilot. **AI Expert**, p. 21-26, December 1993.
- [11] H.J.S. Coutinho, A.B. Tcholakian y L.M. Oliveira. Sistema de apoio a decisão para a minimização do tempo de regata. **Simposio Brasileiro de Pesquisa Operacional (SBPO)**, Juiz de Fora-MG, Brasil: UFJF, 1999.
- [12] S. Mangano. Algorithms for directed graphs: A unique approach using genetic algorithms. **Dr. Dobb's**, April 1994.