Drawing Database Schemas with DBdraw *

Giuseppe Di Battista, Walter Didimo,
Maurizio Patrignani, and Maurizio Pizzonia
{gdb,didimo,patrigna,pizzonia}@dia.uniroma3.it

Dipartimento di Informatica e Automazione, Università di Roma Tre,
Via della Vasca Navale 79, 00146 Roma, Italy.

A wide number of practical applications would benefit from automatically generated graphical representations of database schemas. A well accepted standard for drawing such schemas represents tables by boxes, and table attributes correspond to distinct stripes inside each table. Links, connecting attributes of two different tables, represent referential constraints or join relationships, and may attach arbitrarily to the left or to the right side of the stripes representing the attributes. The DBdraw software tool allows the user to automatically produce a drawing of a database schema according to the drawing standard sketched above.

From an architectural point of view DBdraw takes advantage of the capability provided by the Windows® operating system for importing the database schema and producing an output that may be rapidly inserted into database documentation. The user specifies a database, by now only Microsoft® Access databases (.mdb files) are supported. DBdraw extracts the schema from the database, computes the drawing and inserts it in a Microsoft® Word document whose name is again specified by the user. The drawing is stored in a vectorial format and may be further modified by the user. Figure 1 shows a snapshot of a DBdraw session.

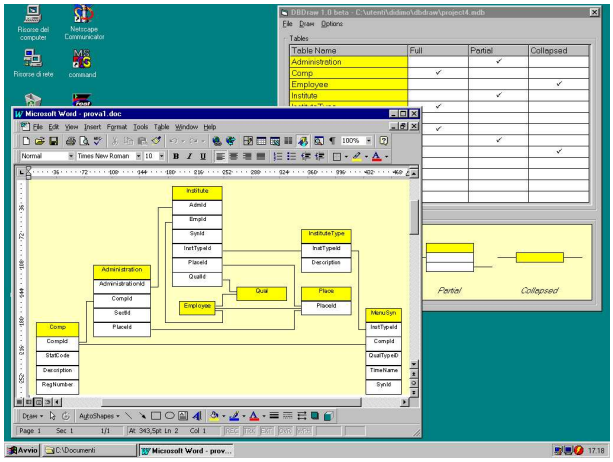


Figure 1: A snapshot of a DBdraw session. On the front the drawing of a database schema embedded in a Word document. On the back the main DBdraw window that allows to select the tables to be shown.

Figure 2 sketches the architecture of DBdraw. The core of the system is a drawing engine which is based on GDToolkit [2] and LEDA [3]. The drawing technique used by the drawing engine is a suitably modified topology-shape-metric approach. Briefly, it consists of four steps:

Constrained Planarization A planarization is performed on the graph underlying the database schema. The purpose of this step is to obtain a planar embedding such that the order of the links around each table is consistent with the specific sequence of attributes of the table. Dummy vertices of degree four are introduced to replace crossings, see Figure 3(a).

U-Turns Assignment This step deals with the left-to-right development of the drawing. From this perspective the link may monotonically follow in the left-to-right direction or may perform one or more u-turns. A (possibly empty) sequence of u-turns is associated with each link trying to minimize their total number, see Figure 3(b).

*Work partially supported by: "Progetto Algoritmi per Grandi Insiemi di Dati: Scienza e Ingegneria", MURST Programmi di Ricerca di Rilevante Interesse Nazionale.

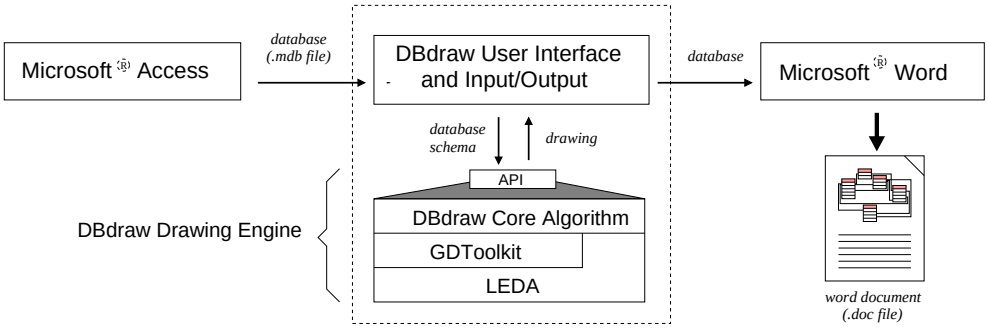
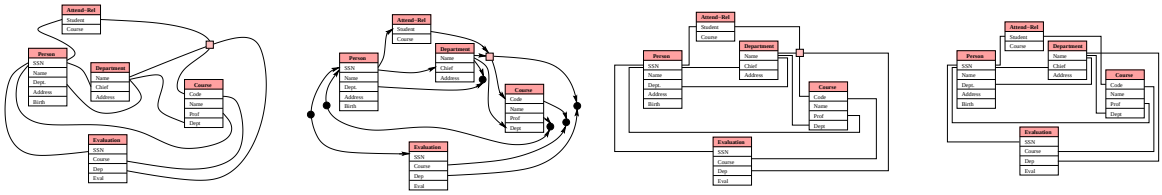


Figure 2: The architecture of DBdraw and the flow of data between the systems.



(a) Constrained planarization (b) U-turn assignment (c) Orthogonalization (d) Constrained compaction

Figure 3: Illustration of the main steps of the DBdraw algorithm.

Orthogonalization This step associates an orthogonal shape to the schema. Each u-turn is replaced with two bends of 90 degrees. The shape is such that links approach tables horizontally. Coordinates of bends and tables are still unspecified, see Figure 3(c).

Constrained Compaction The output of this step is a complete drawing of the database schema. The length of the link and the size of the vertices are computed, keeping as small as possible the area and the total link length. The adopted technique allows us to exactly specify the incidence point of each link on the boxes representing the tables involved in the link. Dummy vertices introduced in the Constrained Planarization step are removed, see Figure 3(d).

The details of the algorithm are given in [1]. The DBdraw tool is available on the Internet at

`http://www.dia.uniroma3.it/~dbdraw`

References

[1] G. Di Battista, W. Didimo, M. Patrignani, and M. Pizzonia. Drawing relational schemas. In W. C. de Leeuw and R. van Liere, editors, *Data Visualization 2000*, Eurographics. Springer-Verlag Wien, 2000.

[2] Gdtoolkit 3.0: An object-oriented library for handling and drawing graphs, 1999. Third University of Rome, http://www.dia.uniroma3.it/~gdt.

[3] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, New York, 1998.