Datenbank Spektrum (2012) 12:215–218
DOI 10.1007/s13222-012-0102-x

FACHBEITRAG

CityPlot: Colored ER Diagrams to Visualize Structure and Contents of Databases

Martin Dugas · Gottfried Vossen

Received: 7 December 2011 / Accepted: 3 September 2012 / Published online: 13 September 2012

Abstract CityPlot generates an extended version of a traditional entity-relationship diagram for a database. It is intended to provide a combined view of database structure and contents. The graphical output resembles the metaphor of a city. Data points are visualized according to data type and completeness. An open source reference implementation is available from http://cran.r-project.org/.

Keywords Entity-relationship model · Database contents · Database visualization · Data completeness

1 Introduction

Entity-Relationship (ER) models have long provided an abstract and conceptual representation of data [1]. ER models and derived ER diagrams are commonly used in database modeling [2, 3], due to their intuitive visualization and their independence from implementation details. Moreover, many extensions of the basic ER model have been proposed over the years, and methods as well as tools have been developed to exploit these models in practical database design [3–6]. Generally, ER diagrams show the conceptual schema of a database, but they do not visualize anything about the actual data points (records, tuples, objects) of a database in terms of the number of entries or completeness of data. The goal of this paper is to change this.

During the software life cycle databases can change a lot over time: data records are inserted, updated and deleted; software updates can come along with major changes to the structure of the underlying database, e.g., involving new attributes and entities (and possibly new relationships). In addition, databases in various application domains can be very complex: 100+ attributes and 100.000+ records per single table are common in real-world databases; therefore methods and tools to provide an overview for such systems are needed.

From a data analysis perspective, identification of data points suitable for further processing steps is of key importance. As a first step, data completeness needs to be assessed [7]. Especially when manual data entry is performed, as, for example, in many clinical applications of databases, the existence of an attribute in the database schema does not guarantee that data is being entered. For user acceptance reasons, enforcing non-NULL attributes is often impossible since not all values may be known when a tuple is inserted. Similar, in many automated data collection systems, the underlying schema contains considerably more attributes than are actually assigned values by, say, sensors or measuring instruments.

Secondly, data types are important: Indeed, available statistical methods for continuous numeric attributes are quite different from procedures for categorical attributes [8]. Date/time attributes and free-text items need completely different data processing techniques. For instance, natural language processing (NLP) methods can be applied to free-text attributes. To plan data analysis activities, it is important to get an overview: How many attributes of what type and how many records per attribute are available? This can be challenging given the high number of attributes and records.

M. Dugas (✉)
Institute of Medical Informatics, University of Münster, Albert-Schweitzer-Campus 1, Gebäude A11, 48149 Münster, Germany
e-mail: dugas@uni-muenster.de

G. Vossen
Department of Information Systems, University of Münster, Leonardo-Campus 3, 48149 Münster, Germany
e-mail: vossen@uni-muenster.de

Visualization [9] can help to tackle these issues, in particular when properly combined with ER modeling. To this end, we present an extended version of traditional ER diagrams in the following which meets the following design goals:

- Comprehensive visualization of database structure regarding entities and their relationships,
- information regarding the number of records and the number of attributes per database entity,
- information about available data types: proportion of numeric, categorical, date/time and other data points per entity,
- identification of incomplete data points.

2 CityPlot

To visualize database structure regarding entities and their relationships (with cardinality types $1:1$, $1:n$, or $n:m$), standard ER diagrams [1, 2] are used as a starting point, with rectangles representing entity types and diamonds representing relationships. The size of an entity box is adjusted according to available database contents for this entity: the width of the box corresponds to the number of attributes, i.e., wide boxes represent entities with many attributes. The height of an entity box is adjusted according to the number of database records for this entity, i.e., high boxes represent entities with many records.

To visualize some given database contents, each available data point (tuple, record) is represented by a colored small spot within the corresponding entity box. Missing values are depicted as empty regions. Each column of spots corresponds to a certain attribute and is colored according to its data type (i.e., numeric, categorical, date/time, other). This enables an inspection of available data types as well as of dataset completeness. "Heat maps" are related visualization techniques which are commonly used in bioinformatics [10].

The graphical output of this extended ER diagram resembles the metaphor of a city. For this reason, this visualization tool is called CityPlot. Each entity corresponds to a building of a city. A high building represents a dataset with many records, a wide building visualizes an entity with many attributes. The relationships between entities correspond to the underground distribution network of the city, therefore these relationships are depicted underneath the respective entity boxes.

3 Results

An open source reference implementation of CityPlot with sample datasets, written in R [11], is available as package

d1.csv	d2.csv	1:1
d2.csv	d4.csv	$n:m$
d1.csv	d3.csv	$1:n$
d3.csv	d5.csv	$n:1$

Fig. 1 Control file for CityPlot. *Each line* describes a relation between two database entities. Each csv file contains all data points for a single entity

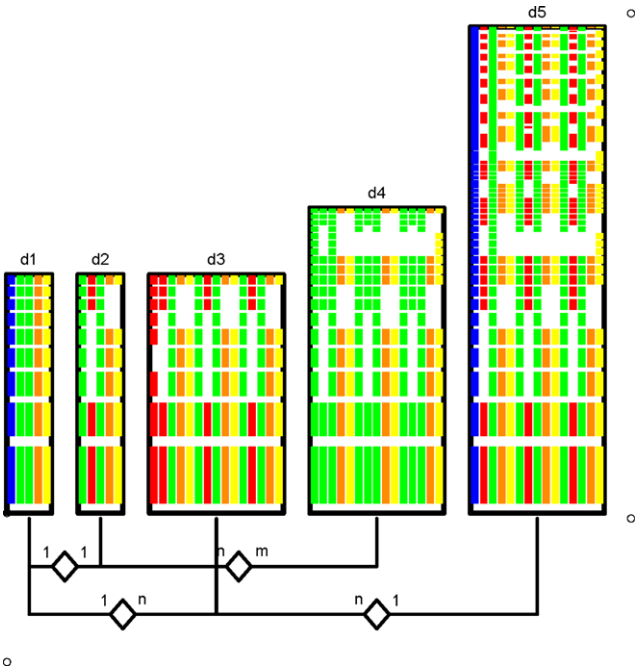

Fig. 2 CityPlot for a database with five entity types, 5 to 15 attributes each and 9 to 108 records per entity type. Numeric data points are colored in *blue*, categorical ones in *green*, date/time ones in *yellow* (time: *light yellow*, date: *dark yellow*), others in *red*. White spots correspond to missing data (Color figure online)

CityPlot from http://cran.r-project.org/. This implementation requires as input a csv export file for each database entity and a control file (for an example see Fig. 1) to describe relationships between various database entities.

A heuristic procedure is applied to automatically assign data types to data columns in csv files: First, matching patterns regarding time and date strings are checked. If there is no match, data is checked for numeric values. Categorical values can also be represented by numbers; therefore a heuristic cutoff criterion regarding the quantity of different levels is applied, in order to differentiate between numeric and categorical values. A similar criterion is used to distinguish between text-based categorical items and other data types. This procedure to assign data types is required, because in some real-world databases attribute definitions are not precise; for instance, text attributes can be used to store numeric data.

To enable simultaneous assessment of entities with small and very large number of entries, the height of entity boxes is

plotted on a logarithmic scale. Figure 2 presents an example of a CityPlot for a database with five entity types.

Figure 3 presents a CityPlot of the Heart Disease Data Set from UCI Machine Learning Repository [12]. This data set has been extensively studied in more than 50 publications. It demonstrates that CityPlot can be applied to real databases of considerable size. For instance, it can be seen from this figure that some attributes have many missing values and that the distribution of these missing values is very probably not random.

ENSEMBL [13] is a database—much larger than the previous example—of genome resources for more than 60 species with a particular focus on human genome data. This database is available for public download and consists of 200+ tables. Figure 4 shows a CityPlot for a subset of ENSEMBL. Due to the logarithmic scale of the entity height, tables with few (less than 100) and many (100.000+) tuples can be displayed on the same plot. Again, it can be seen that there are non-random patterns of available and missing data. Entities with lots of numeric and categorical data can be identified easily.

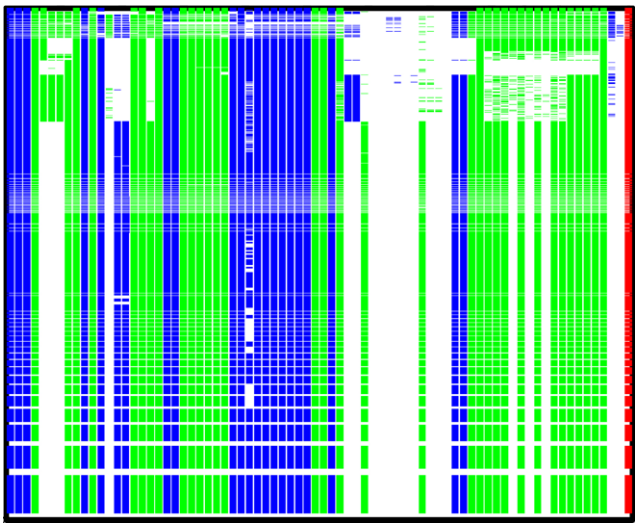

Fig. 3 CityPlot Heart Disease data set (from UCI machine learning repository). It consists of 75 attributes, most of which are categorical (*green spots*) and numeric (*blue spots*). Some attributes show data values in almost all tuples, while for other attributes the proportion of missing values is very high (*white spots*) (Color figure online)

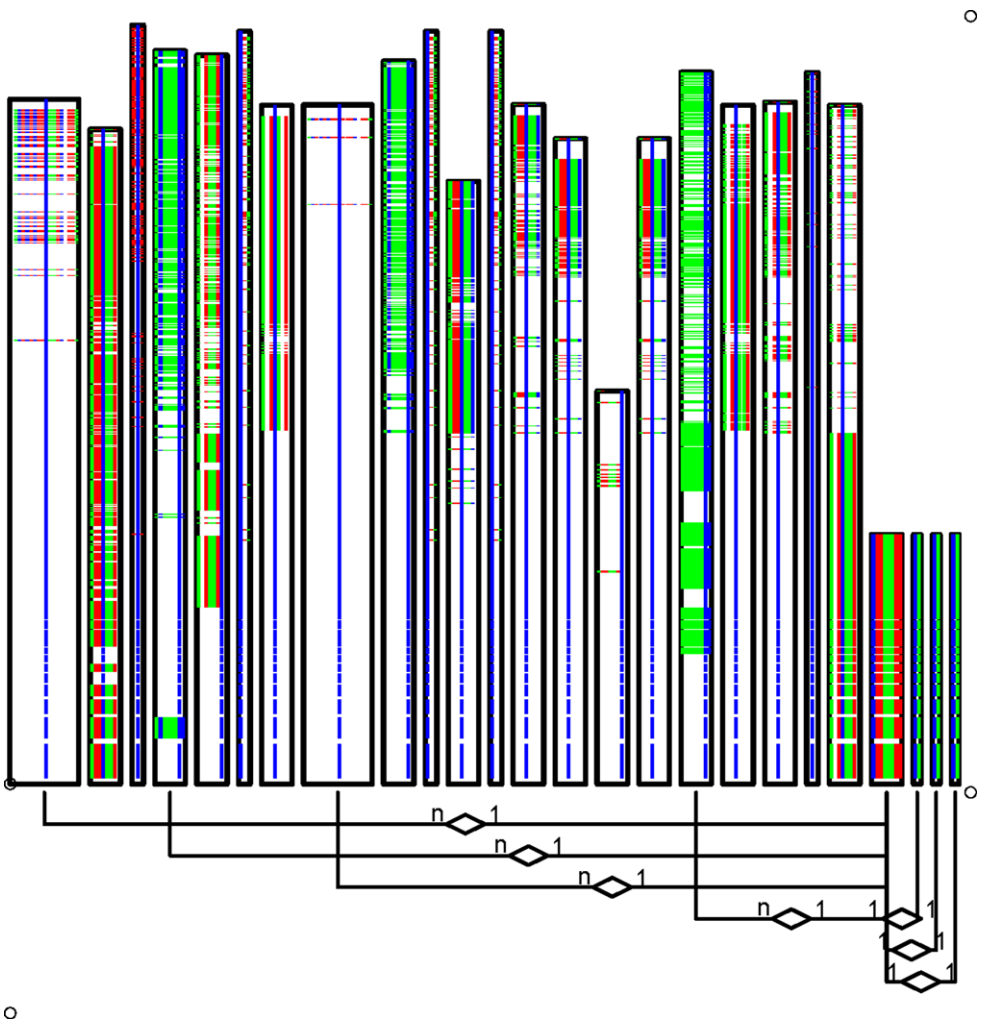

Fig. 4 CityPlot for a subset of the ENSEMBL database. 25 tables with 3 to 18 attributes and 54 to 196.971 tuples are presented. Most attributes are categorical (*green spots*). There are many missing values (*white spots*) and the pattern of missing data appears to be non-random. The information about relationships between entities is available only for a subset of the entities (Color figure online)

4 Discussion

Real-world databases, such as electronic health record or enterprise resource planning systems, can be very large and complex. Therefore it can be a challenging task to obtain an overview and, more specifically, to identify suitable data subsets for further analysis. For the management of information systems it is essential to understand the overall structure of the underlying database and to know the strengths and weaknesses of its contents regarding data analysis.

Interestingly, data profiling methods, which have also been developed with the goal in mind to gather or summarize information on a given database, its contents as well as its schema, have so far vastly neglected the possibility to visualize database structure and contents in an intuitive way; CityPlot tries to fill this gap.

Many real-world databases evolve over time without systematic modeling: tables are added to provide or support new functionality—sometimes resulting in hundreds or even thousands of tables per system—and information about relationships between the entities therein is sometimes not available any more (see Fig. 4). Therefore, data profiling methods with visualization to obtain an overview are warranted.

In contrast to information visualization systems like Polaris [14], which has been commercialized by Tableau Software [15], CityPlot is a non-interactive method. For this reason, our approach is not a typical InfoVis [16] procedure. The advantage of a non-interactive method is that for a given database setup there is only one plot. Thus, comparison of different databases or different versions of the same database becomes easier. On the other hand, by interactive systems a much deeper exploration of the data is possible to validate discoveries. CityPlot is a pixel-oriented visualization technique, i.e., each attribute value of the data is mapped to a single colored pixel. To account for very large number of records a logarithmic scale is applied.

CityPlot enables to spot missing values. In principle, database attributes could be defined as "NOT NULL," but in real systems it is very common that incomplete data collections need to be stored, therefore it is left to the application logic and the users to provide complete data. From a data analysis perspective, it is important to understand patterns of missing values to avoid bias.

Previous work [17] has described a method to visualize structure and quality of data, specifically applicable to medical databases. The new approach is more generic and—due to the city metaphor—more consistent and easier to comprehend. Indeed, the proposed extended ER diagram tries to integrate a visualization of database structure and its contents. CityPlot can be used to generate high-level snapshots of a database over a period of time. By this means it can be shown how the system is changing—both by additional data points and by updates of the database structure.

Acknowledgements This work was supported by Deutsche Forschungsgemeinschaft (DFG) grant DU 352/5-1 and Bundesministerium für Bildung und Forschung (BMBF) grant 01EZ0941A. The authors thank the reviewers for their constructive comments.

References

1. Chen P (1976) The entity-relationship model—toward a unified view of data. ACM Trans Database Syst 1(1):9–36
2. Elmasri RA, Navathe SB (2010) Fundamentals of database systems, 6th edn. Pearson Addison-Wesley, Boston
3. Silberschatz A, Korth HF, Sudarshan S (2010) Database system concepts, 6th edn. McGraw-Hill, New York
4. Batini C, Ceri S, Navathe SB (1992) Conceptual database design—an entity-relationship-approach. Benjamin/Cummings, Redwood City
5. Markowitz VM, Shoshani A (1992) Representing extended Entity-Relationship structures in relational databases: a modular approach. ACM Trans Database Syst 17(3):423–464
6. Fahrner C, Vossen G (1995) A survey of database design transformations based on the entity-relationship model. Data Knowl Eng 15(3):213–250
7. Naumann F, Freytag JCh, Leser U (2004) Completeness of information sources. Inf Syst 29(7):583–615
8. Altman DG (1991) Practical statistics for medical research. Chapman & Hall, London
9. Keim D, Kohlhammer J, Ellis G, Mansmann F (2010) Mastering the information age—solving problems with visual analytics. The Eurographics Association, Goslar
10. Wilkinson L, Friendly M (2009) The history of the cluster heat map. Am Stat 63(2):179–184
11. R Development Core Team (2009) R—a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna
12. Heart disease data set. UCI machine learning repository. http://archive.ics.uci.edu/ml/datasets/Heart+Disease. Accessed May 2, 2012
13. Flicek P, Ridwan Amode M, Barrell D et al (2012) Ensembl 2012. Nucleic Acids Res 40:D84–D90
14. Stolte Ch, Tang D, Hanrahan P (2002) Polaris: a system for query, analysis, and visualization of multidimensional relational databases. IEEE Trans Vis Comput Graph 8(1):1–14
15. Tableau Software Inc. http://www.tableausoftware.com/. Accessed May 4, 2012
16. infovis Wiki: information visualizaton. http://www.infovis-wiki.net/index.php?title=Information_Visualization. Accessed May 4, 2012
17. Dugas M, Kuhn K, Kaiser N, Überla K (2001) XML-based visualization of design and completeness in medical databases. Med Inform Internet Med 26(4):237–250