

BLer: A Boundary Labeller for Technical Drawings^{*}

Michael A. Bekos, Antonios Symvonis

National Technical University of Athens
School of Applied Mathematical & Physical Sciences
15780 Zografou, Athens, Greece
{ mikebekos | symvonis } @math.ntua.gr

Abstract. In technical drawings and medical drawings/maps it is often common to explain certain features of the drawing by blocks of text that are arranged on its boundary. Bekos et. al. introduced the term *boundary labelling* to describe this special type of labelling and presented algorithms for automatic boundary labelling. **B**Ler is a prototype tool aiming to automate the boundary labelling process. It targets the area of technical and medical drawings and incorporates implementations of boundary labelling algorithms presented in [1].

1 Introduction

One of the most challenging tasks in the map labelling is the automated visualization of the information on a map, i.e. the association of text labels with graphical features. Most applications focus on placing the labels so that: i) they do not overlap with each other, ii) they are close to (if possible, next to) the feature they are associated with, and iii) they do not obscure other features of the drawing. Such a labelling is clearly not achievable in the case of large labels (or, equivalently, dense point sets). Large labels are common in technical drawings and medical atlases where certain features are explained by blocks of text.

In technical drawings, it is quite common to place the labels on the boundary of the drawing and to connect them to the features they describe by non-crossing poly-lines (*leaders*). Bekos et. al. introduced the term *boundary labelling* to describe this special type of labelling and presented algorithms for automatic boundary labelling.

^{*} This work has been partially supported by the EPEAEK program Pythagoras 89181(28).

To the best of our knowledge, no drawing software includes support for automated placement of labels on (or near) the boundary of the drawing. Only SmartDraw (<http://www.smartdraw.com>) seems to provide such labellings, however drawings are produced manually (based on a wide range of templates), not automatically.

BLer is a prototype tool that supports the boundary labelling model and facilitates the annotation of drawings with text labels. It is suitable for the production of medical atlases and technical drawings, where the basic requirement is large labels. It is entirely written in Java based on the yFiles class library (<http://www.yworks.com>).

2 The labelling process

BLer enables the user to quickly generate boundary labellings from scratch. It's environment (see Figure 1) supports multiple views of the labelling and directs the user through the steps of the labelling process. The labelling process followed by the user in order to create a labelling of a diagram consists of the following steps:

1. *Diagram loading.* The user loads the diagram to be annotated with labels. The diagram must be in either the jpeg or the gif format and it is treated as *background*.
2. *Definition of enclosing rectangle.* The user defines a rectangle which contains the drawing. The labels will be placed at the boundary of this rectangle.
3. *Definition of point features.* The user specifies the point features of the diagram. These point features can only be located inside the enclosing rectangle. For each point feature the user specifies the text of the corresponding label.
4. *Production of boundary labelling.* The user has to specify the parameters of the labelling. These include: the sides of the enclosing rectangle where labels can be placed (medical atlases usually have the labels on their left or right sides, technical drawings have labels on all 4 sides), the type of the labels (variable or uniform size), the type of the leaders¹ (*opo*, *po*, or *s*), the type of the *ports*² (fixed or sliding along the label side).

¹ See [1] for the definition of the leader types.

² The point where a leader is connected with its corresponding label is referred to as *port*.

The user can also instruct `BLer` to use a *legend* (see Figure 3). Legends are useful in cases where the labels (due to their size and number) do not all fit on the boundary of the enclosing rectangle. When a legend is used, labels of appropriate uniform size are used, each containing a number which refers to a particular line of the legend. The legend is treated as a floating object and can be manually placed anywhere around the resulting labelling. The labellings produced by the current version are based on algorithms developed by Bekos et. al. [1]. They minimize the total leader length and contain no crossings. The resulting drawings are simple, in terms of readability, ambiguity and legibility.

As expected from a labelling tool, it supports storing, reloading and post-processing of labellings. It provides advanced graphic functionality, including popup menus, printing capabilities, custom-zoom, fit-in window, selection, dragging and resizing of objects.

3 Examples

We present some characteristic outputs of `BLer`. Figure 2 depicts a medical map of the regions of a human head. We have chosen *type-po* leaders, such that labels are placed directly to the right of the drawing and the ports are sliding.

Figures 3 and 4 shows two technical drawings. Such drawings are very common in technical manuals of several devices. Figure 3 depicts a technical drawing of a cb radio, where we have chosen *type-s* leaders, such that labels are treated as legends around the drawing and the ports are sliding. Figure 4 depicts a technical drawing of a motherboard, where we have chosen *type-opo* leaders, such that labels are placed directly to the right of the drawing and the ports are sliding.

4 Conclusions and Future Work

We support a series of models and algorithms for efficient boundary labelling of point sets. We have implemented most of the algorithms presented in [1]. Currently, we are working on incorporating drawings of minimum number of bends and on supporting more flexible forms of defining the point features to be labelled.

References

1. M. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. In Janos Pach, editor, *Proc. 12th Int. Symposium on Graph Drawing (GD'04)*, Lecture Notes in Computer Science, New York, September 2004.



Fig. 1. A snapshot of BLer

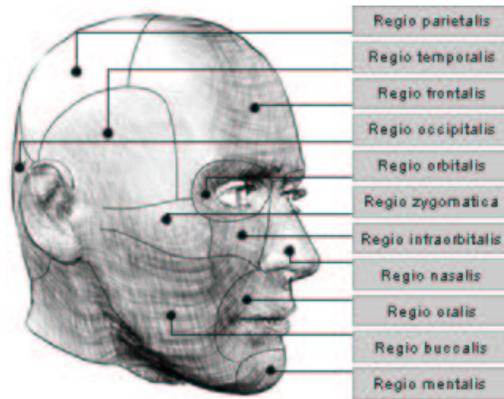


Fig. 2. A medical map of the regions of a human head. Leaders are *po*-type and the labels are placed directly to the right of the drawing.



Fig. 3. A technical drawing of a cb radio. Leaders are *s*-type and the labels are treated as legends.

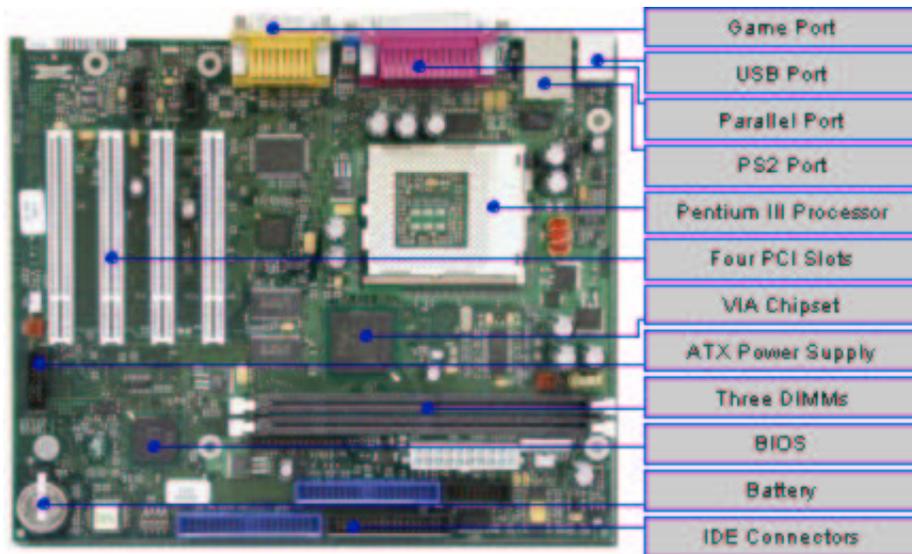


Fig. 4. A technical drawing of a motherboard. Leaders are *opo*-type and the labels are placed directly to the right of the drawing.