INFORMATIONAL DIAGRAMS IN SCIENTIFIC DOCUMENTS

Catherine M. Carriero, Robert P. Futrelle, Nikos Nikolakis and Michael Tselman

Biological Knowledge Laboratory. College of Computer Science 161CN
Northeastern University, 360 Huntington Avenue, Boston, MA 02115 USA

{carrier,futrell,nikos,misha}@corwin.ccs.northeastern.edu
Phone: (617) 437-2076 FAX: (617) 437-5121

Abstract

Much of the work on diagrammatic reasoning assumes that the structure of the diagrams has already been determined. This is not the case when dealing with diagrams in the published scientific literature. So we are building a system that can understand these diagrams, starting from the graphics primitives such as lines and polygons and resulting in symbolic and veridical representations of the diagram structure. To date we have pursued an approach based on Graphics Constraint Grammars which describe diagram components as a hierarchically structured collection satisfying various geometric constraints. The grammars encode the conventional aspects of diagram structure as well as the informational aspects related to perceptual grouping. As part of this, we have developed spatially indexed data structures that allow the direct spatial storage and recovery of diagrammatic and symbolic information.[1]

Introduction

The Biological Knowledge Laboratory is concerned with the analysis of entire biological research papers, text and graphics, i.e., the automated construction of knowledge bases representing the papers which can be accessed by a scientist. This is the context of all the discussions below. There are very cogent arguments as to why we should pay special attention to diagrams [Larkin and Simon, 1987; Schooler and Engstler-Schooler, 1990].

We'll illustrate our approach using two different types of diagrams. The first is a data graph, Figure 1.

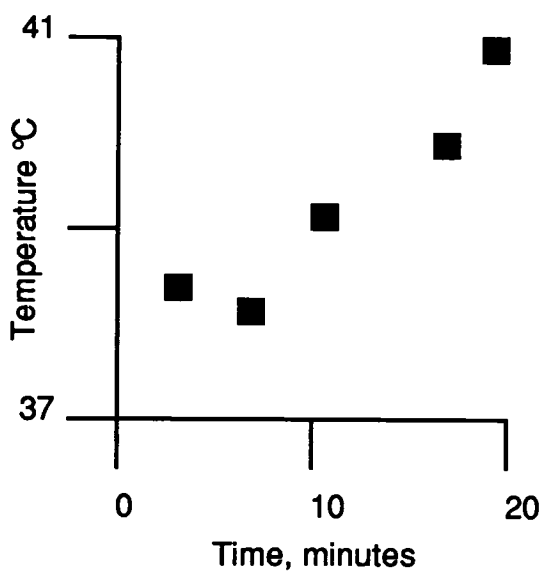

Figure 1. A simple data graph illustrating *information* (the data points) presented on a *substrate* (the quantitative grid defined by the axes).

Diagrams such as this are organized into the *information* component, the five data points, presented on a *substrate* [Futrelle, et al, 1992]. The substrate is the two-dimensional surface parameterized by temperature and time in Figure 1. The data points have no simple relation to one another; after all they were derived from

[1] This work supported in part by grant DIR88-14522 from the National Science Foundation to RPF.

experiment and their values are not under the direct control of the experimenter. As such, the data points have a high information content. The scale line structures, in contrast, are created to have a simple structure with elements arranged in an orderly way. They have a lower information content.

Besides those aspects of the diagram captured in the *information / substrate* distinction, the other major ingredient of diagram structure is the set of *design conventions* in force at any time. This description of diagrams parallels distinctions in natural language in which *information* plays a part, e.g., in the relation between word length and frequency, along with the *conventions* of any particular language at some point in time, e.g., its lexicon and grammar.

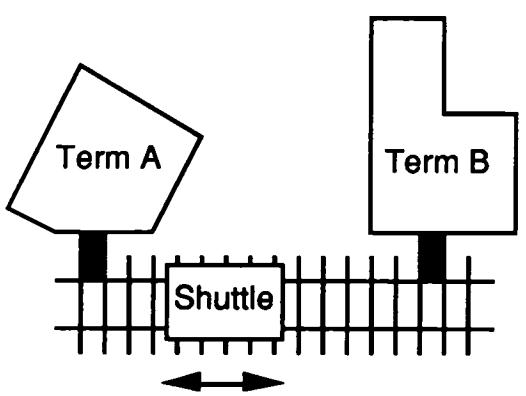

Figure 2. A veridical diagram showing rail shuttle service from one terminal building to another. The shuttle *functions* include the ability to move on the tracks and to take on, transport and discharge passengers and their baggage.

Figure 2 is more complex. There is not such a clear distinction between information and substrate. The best candidates for substrate elements are the tracks and especially the ties, the short vertical lines connecting the two tracks. In the final analysis, the question of whether an item is information or substrate may depend on the use to which the diagram analysis is to be put. E.g., the exact shape of the terminal building may matter in some cases and may not in others.

Graphics Constraint Grammars

A Graphics Constraint Grammar is a collection of productions of the form,

```
LHS ==>  RHS1, RHS2,....
```

The constituents on the right hand side may be simple geometrical objects such as points, lines, polygons or text, or they may be higher level objects which appear as the left-hand-sides of other grammar rules. In addition, each rule contains *constraints* that must hold between the right-hand-side constituents [Helm, Marriott and Odersky, 1990]. The constraints are typically type constraints, e.g., `(Linep RHS1)`, which says that `RHS1` must be a line, or more general geometric constraints, e.g., `(Parallel RHS1 RHS2)` which states that the two constituents must be parallel to one another.

In addition, there are *propagation rules* which specify how the properties (attribute-value pairs) of the left-hand-side are to be computed from the properties of the right-hand-side constituents. Chief among the higher-level properties is the *bounding_box* which gives a rough characterization of the size, shape and location of the left-hand-side element. For example, in Figure 1, the bounding box for *X_scale* would be computed as the smallest rectangle enclosing the horizontal scale line, the tick marks, their numerical labels and the title, "Time, minutes." This rectangle would be propagated to *X_scale* as the value of the *bounding_box* attribute.

Perceptual grouping and Generalized Equivalence Relations

We've pointed out that the substrate of a diagram contains orderly arrangements of

simple items. When viewed they tend to come together in perceptual groupings. Most of the known perceptual grouping processes seem to be closely tied to *equivalence relations*. Objects which tend to group perceptually share the same value of some simple geometrical property, and are equivalent in that sense. The tick marks in Figure 1 and the railroad ties in Figure 2 group together because they are aligned and equally spaced; these are equivalence relations on the set of lines.

Clearly, one of the most fundamental geometrical equivalence relations is *coincidence*. If two objects are in the same place, they tend to be grouped perceptually. A weakened form of coincidence is *near*. *Near* is what we call a *Generalized Equivalence Relation*. It expresses the proximity of objects. *Near* is reflexive and symmetric, but not transitive. The nearer two items are, the more they tend to group perceptually.

The grammars we write depend heavily on Generalized Equivalence Relations. They serve to group objects at one level into higher level objects at the next level.

Spatial Indexing - Graphic and Symbolic

Conceptually, proximity is a powerful notion. In much of the physical world, interactions require proximity or even direct physical contact. Therefore our analysis system uses data structures that store diagram constituents, both graphics primitives and higher-level objects, in a *spatially indexed form* [Samet, 1990; Futrelle, 1990]. The technique is very simple and is shown in Figure 3. The grid is much less detailed than a bitmap. Typically a cell in the grid represents about a 1x1 mm area of a published diagram. Each cell in the x,y spatial array contains a reference to any object which touches it or covers it. In Figure 3 there are 9 references to the circle, 6 to the square and 7 to the triangle. Each object contains a reference to every cell it occupies, so once an object is discovered, all objects near to it can be discovered quickly.

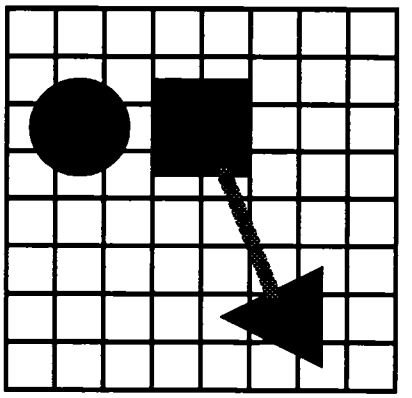

Figure 3. Spatial indexing of three geometrical objects. A reference to each object is stored in every spatial cell which the object touches. To find objects *near* to the circle, the cells near the circle's cells are checked and a reference to the square is found. The greyed line is a *symbolic link* between the square and the triangle which can be stored as a geometric object for direct spatial access.

The spatial indexing notion is very powerful and can be used to directly store symbolic information also. Let's assume in Figure 3 that the square and the triangle are related by the *Same_owner* relation. We could easily place this information in symbolic data structures attached to both objects or indexed in an *Owner* structure. But we could also place an actual physical link into the spatial array, an object of type *Same_owner* with geometrical representation *line* as shown in Figure 3.

The power of such an approach is that it directly implements the "spatial storage of information" paradigm alluded to in Larkin and Simon [1987]. Information about objects is literally stored at the objects and relations literally connect objects. So if the system is looking at a certain small set of cells it can discover relations about them that are stored there. Furthermore, it can literally

traverse the relation line to find related objects. This gives the entire computation an *animate vision* flavor [Ullman, 1986; Futrelle, 1990].

Semantics, functions and reasoning

In our work to date, we have pursued an approach that has been generally successful in natural, language analysis – the separation of syntax from semantics. The initial parsing focuses on purely geometrical relations (syntax) without concerning itself with semantics. This is particularly appropriate for diagrams because low-level graphical elements do not have the arbitrary, yet precise meanings that words have in natural language. "Cat" has some serious semantics attached to it, but the graphical item "[redacted]" does not. Its interpretation is highly context dependent – it could be two cars of a train, two resistors in a circuit drawing, or two related tribes in an anthropological discussion. There is no easy solution to this interpretation problem. The subject area of a paper, the figure caption, the references to the figure in the text, the conventions of the field – all contribute to the semantic interpretation of a diagram.

Our work to date has emphasized static descriptions, the parse trees and semantic frames that result from diagram analysis. But we are looking toward analysis techniques that include the *function* of the items being analyzed. For example, in Figure 2, the fact that a train *moves* on tracks could assist the analysis by decoupling the rectangle from the track and ties substrate and identifying it as the train. The functionality attached to objects can assist the process of reasoning about them.

Acknowledgements

The authors wish to thank Art Miller and others in the BKL for help on the manuscript.

References

[Futrelle, 1990] R.P. Futrelle, "Strategies for Diagram Understanding: Object/ Spatial Data Structures, Animate Vision, and Generalized Equivalence," in *Proceedings of the 10th ICPR*, IEEE Press, 1990, pp. 403-408.

[Futrelle, et al, 1992] R. P. Futrelle, et al, "Understanding Diagrams in Technical Documents", *IEEE Computer*, June, 1992, to appear.

[Helm, Marriott and Odersky, 1990] R. Helm, K. Marriott, and M. Odersky, "Building Visual Language Parsers," *CHI '91* , ACM, New York, 1991, pp. 105-112.

[Larkin and Simon, 1987] J. H. Larkin and H. A. Simon, "Why a diagram is (sometimes) worth ten thousand words," *Cognitive Sci.*, Vol 11, 1987, pp. 65-99.

[Samet, 1990] H. Samet, *The design and analysis of spatial data structures*, (2 volumes), Addison-Wesley, Reading, MA, 1990.

[Schooler and Engstler-Schooler, 1990] J.W. Schooler and T.Y. Engstler-Schooler, "Verbal overshadowing of visual memories: Some things are better left unsaid," *Cognitive Sci.*, Vol. 22, 1990, pp. 36-71.

[Ullman, 1986] S. Ullman, "Visual routines: Where bottom-up and top-down processing meet," in *Pattern Recognition by Humans and Machines: Visual Perception*, E.C. Schwab and H.C. Nusbaum eds., Vol. 2, Academic Press, San Diego, CA, 1986, pp. 159-218.