**Demonstration of the**

**Diagram Understanding System (abstract)**

Robert P. Futrelle

Biological Knowledge Laboratory
College of Computer Science
Northeastern University, 161CN
Boston, Massachusetts 02115
futrelle@ccs.neu.edu, http://www.ccs.neu.edu/home/futrelle

The Diagram Understanding System is designed to parse a variety of classes of complex diagrams, each class specified by a grammar. All diagrams are assumed to be in vector form, made up of primitives such as lines, rectangles, Bézier curves, and text. The grammars are context-based constraint grammars. The constraints are the conventional spatial constraints such as *aligned*, *near*, *above*, as well as qualifiers such as *short* and *long*. Parsing proceeds top-down. Each rule that is activated during the parse is given a limited set of elements within which it attempts to discover the appropriate objects satisfying the constraints. The limited set of elements given each rule is the "context" of that rule. The successive narrowing of the context as the parse is pursued in depth is one of the reasons for the efficiency of the system.

The diagram collection used in this work consists of 630 diagrams faithfully redrawn from the scientific research literature, almost exclusively from the biology literature. These diagrams are typically complex, comprised of between 100 and 200 vector primitives. The major grammars developed so far cover a subset of the diagrams in the collection, including x,y data graphs and linear gene diagrams. In addition, we have developed grammars for finite-state automata diagrams. One aspect of parsing diagrams that makes them somewhat different from natural language parsing is the sharing or re-use of objects. This is most evident in the classic example of the dividing wall in a diagram of a building -- the wall is simultaneously a component of two distinct room descriptions. In x,y data graphs that have multiple subgraphs it is common for scale lines and their tick marks, as well as axis labels to apply to more than one graph (to the right or above). The parser creates objects at every level, from the primitive vector elements to higher-order entities in the parse. The constraint satisfaction computations allow re-use of any of these objects as necessary.

In diagram parsing, especially during system development and testing, it is important to "see" the structure of the parse. To do this we have developed the DUS Inspector (DUSI) that maintains a correspondence between the parse structure and the displayed image. Thus we can ask to have the elements contained in a particular parse node highlighted in the graphics window, and conversely, we can select an item in the graphics window and find what parse objects contain it.

The system was developed in Macintosh Common Lisp, with heavy use of CLOS. Grammars are processed to produce CLOS classes corresponding to every non-primitive (left-hand side), so the resulting parse structures are object graphs. The most thorough tests have been done on quite slow machines, and even then ran well under a minute to parse diagrams of 100 to 200 primitives. On current machines at the demonstration, we estimate that typical parses will run in ten seconds or less. (Since there are 32M seconds per year, that rate exceeds the rate of production of diagrams in the entire biomedical literature.)

The non-interactive (non-DUSI) portions of the code are being ported to Unix.

In the demonstration, participants will be able to experiment with grammars applied to a variety of diagrams as well as inspect solutions and discuss the implementation.

**References**
