

of responsibility for managing the display, Emacs greatly simplifies the task of writing correct extensions.

*Emacs updates the display only when waiting for user input*

A single command may apply any number of buffer manipulation primitives, affecting arbitrary portions of the buffer—or perhaps of many different buffers. Similarly, the user may invoke a keyboard macro that carries out a long series of prerecorded commands. In cases like these, rather than showing the intermediate states of the buffer flickering by as it works, Emacs puts off updating the display until it needs to pause for user input, and refreshes the display in a single step at that point. This relieves Lisp programmers from the temptation to optimize or arrange their uses of buffer and window primitives to get smoother display behavior. The simplest code will display well.

By updating the display to reflect the effects of editing primitives automatically and efficiently, Emacs drastically simplifies the work of Lisp authors. Some other systems have picked up on this behavior: most notably, JavaScript programs embedded in web pages can edit the page's contents, and the browser will update the display accordingly. However, a surprising number of systems that invite extensions require careful cooperation between the extension code and the display update code, laying this burden on the extension author's back.

When Emacs was first written, its audience used computers via standalone terminals connected to the computer over serial lines or modems. These connections often were not terribly fast, so a text editor's ability to find the optimal series of drawing operations sufficient to update the screen could have a big impact on the editor's usability. Emacs went to a great deal of trouble to do well here, having a hierarchy of update strategies ranging from the quick-but-limited to the more-work-but-exhaustive. The latter used the same dynamic programming techniques employed by file comparison programs such as *diff* to find the smallest set of operations sufficient to transform the old screen into the new screen. In modern times, although Emacs still uses these algorithms to minimize update work, most of this effort is wasted, as faster processors and faster links between the computer and the display allow simpler algorithms to perform perfectly well.

## **The Controller: Emacs Lisp**

The heart of Emacs is its implementation of its own dialect of Lisp. In Emacs's realization of the Model-View-Controller pattern, Lisp code dominates the Controller: almost every command you invoke, whether from the keyboard, a menu, or by name, is a Lisp function. Emacs Lisp is the key to Emacs's ability to successfully accommodate the wide range of functionality that Emacs has grown to offer.