The \texttt{lattice} package is built on top of \texttt{grid} and provides a quite sophisticated example of writing high-level plotting functions using \texttt{grid}. Because \texttt{lattice} consists of \texttt{grid} calls, it is possible to both add \texttt{grid} output to \texttt{lattice} output, and \texttt{lattice} output to \texttt{grid} output.

\begin{verbatim}
> library(grid)

Adding grid to lattice

Panel functions in \texttt{lattice} can include \texttt{grid} calls. The following example adds a horizontal line at 0 to a standard \texttt{xypplot} (see Figure 1):

\begin{verbatim}
> xyplot(y ~ x | g, panel = function(x, y) {
+   panel.xyplot(x, y);
+   grid.lines(unit(c(0, 1), "npc"), unit(0, "native"),
+               gp = gpar(col = "grey"))
+ })
\end{verbatim}

The following example writes a left-justified label in each strip (see Figure 2):

\begin{verbatim}
> xyplot(y ~ x | g, strip = function(which.given, which.panel, ...) {
+   grid.rect()
+   grid.text(paste("Variable ", which.given, ": Level ",
+                  which.panel[which.given], sep = ""),
+             unit(1, "mm"), .5, just = "left")
+ })
\end{verbatim}

Adding lattice to grid

It is also possible to use a \texttt{lattice} plot as an element of a \texttt{grid} image. The following example splits up the page so that there is an \texttt{xypplot} beside a panel of text (see Figure 3). First of all, the lattice plot is created, but not drawn. \texttt{grid} is used to create some regions and the lattice plot is drawn into one of those regions.

\begin{verbatim}
> someText <- paste("A panel of text", "produced using", "raw grid code",
+                   "that could be used", "to describe",
+                   unit(1, "mm"), .5, just = "left")
\end{verbatim}
Figure 1: A **lattice** panel function using **grid**.

Figure 2: A **lattice** strip function using **grid**.
A panel of text produced using raw grid code that could be used to describe the plot to the right.

Figure 3: A `lattice` plot used as a component of a larger `grid` image.

```r
+ "the plot", "to the right.", sep = "\n")
> latticePlot <- xyplot(y ~ x | g, layout = c(2, 4))
> grid.rect(gp = gpar(lty = "dashed"))
> pushViewport(viewport(layout = grid.layout(1, 2,
+       widths = unit.c(unit(1, "strwidth", someText) +
+                       unit(2, "cm"),
+                       unit(1, "null")))))
> pushViewport(viewport(layout.pos.col = 1))
> grid.rect(gp = gpar(fill = "light grey"))
> grid.text(someText,
+         x = unit(1, "cm"), y = unit(1, "npc") - unit(1, "inches"),
+         just = c("left", "top"))
> popViewport()
> pushViewport(viewport(layout.pos.col = 2))
> print(latticePlot, newpage = FALSE)
> popViewport(2)
```