mation relating to class loaders and the Java™ programming language can be found in "The Java™ Virtual Machine Specification" by Tim Lindholm and Frank Yellin, Addison Wesley, 1997, which is incorporated by reference herein.

Display processor **129** responds to keyboard applet **114** and displays symbols in accordance with keyboard layout **115** on display elements **132**. Keyboard layout information may be stored in NVRAM **128** or similar memory storage while the system is powered down or in a suspended processing state.

AN EXEMPLARY KEYBOARD

FIG. **2** is a block diagram representation of a keyboard input device designed consistent with one implementation of the present invention. Keyboard input device **110** includes a keyboard layout indicator switch **212**, a rotatable keyboard layout selector **221**, a function key bank **214**, application keys **222**, and general keys **224**. Unless indicated otherwise, function key bank **214**, application keys **222**, and general keys **224** are collectively referred to as the keys of keyboard input device **110**. Each key implements a selectable keyboard display element **132** consistent with implementations of the present invention.

The user can use rotatable keyboard layout selector **221** and switching keyboard layout indicator switch **212** in combination to load different keyboard layouts into keyboard input device **110**. In addition to different languages, the keyboard layouts may display special functions associated with certain software applications to be displayed on the keys. For example, spreadsheet functions used with a spreadsheet application may be loaded into function key bank **214**. Descriptive symbols or text can also be displayed on each of the keys in function key bank **214**. Similarly, application keys **222** may display labels associated with macros or other functions often used in a particular application. General keys **224** can also be modified to display symbols or text in a similar manner.

Keyboard layout switch **212** allows a user to select from a set of keyboard layouts as indicated by the corresponding keyboard layout indicators **216–220**. Keyboard layout indicators **216–220** are small display components that present different keyboard layouts available to the user on keyboard input device **110**. For example, layout indicators **216–220** can contain the words "English", "French", or "German". This would indicate that English, French, and German keyboards can be selected by sliding keyboard layout indicator switch **212** accordingly. Although only three layout indicators are described, alternate implementations may include more or fewer indicators.

Rotatable keyboard layout selector **221** enables a user to scroll through many different keyboard layouts. A wheel or similar component may be used as the rotatable keyboard layout selector **221**. As the user rotates selector **221**, different labels identifying available keyboard layouts are displayed in layout indicators **216–220**. In addition, as the user scans the keyboard layouts, the keys may also display the corresponding set of symbols associated with each keyboard layout. The user can then select one of the keyboard layouts provided in layout indicators **216–220** using a predetermined keystroke, a mouse button, or by actuating a touch sensitive overlay on the corresponding layout indicator **216–220**.

FIG. **2** also illustrates a side view of one selectable keyboard display element **132**. A keycap **227** specially designed to display a symbol is attached to the upper portion of selectable keyboard display elements **132**. Keycap **227** includes a translucent protective layer **226** adhered to a display element **228** and inserted in a support structure **230**. Keycap **227** accepts a power input **232** to drive display element **228** and a signal input **234** with a sufficient number of bits to generate the various displayable symbols on keycap **227**. An output signal **236** corresponding to the symbol displayed on keycap **227** is generated when selectable keyboard display elements **132** is actuated. Selectable keyboard display element **132** may be an electromechanical or solid state device, depending on the particular implementation.

EXEMPLARY DYNAMIC CONFIGURATION OF KEYBOARD INPUT DEVICE

FIG. **3** is a flowchart of the steps for dynamically configuring a keyboard layout in a manner consistent with the present invention. Keyboard applet **114** on keyboard input device **110** checks computer system **101** for a layout flag (step **302**). The layout flag is an indicator used to determine a selected keyboard layout to be loaded. Typically, the layout flag is set within the execution environment as an environment variable or within the application as an application variable. This flag permits application **112** to override switch settings on keyboard input device **110**. For example, a software application may temporarily load a different keyboard layout regardless of the settings of rotatable keyboard layout selector **221** and keyboard layout switch **212**. To do this, the software application would set the layout flag to the desired value. Keyboard applet **114** then uses the setting in the layout flag to select the keyboard layout.

Keyboard applet **114** determines if the selected keyboard layout from computer system **101** is already being displayed on the keyboard (step **304**). When the selected keyboard layout is already displayed, keyboard applet **114** uses keyboard layout **115** in memory **126** to map the proper symbol or symbols to each keycap **227** on keyboard input device **110** (step **306**).

Alternatively, keyboard layouts can be downloaded over a network if the selected keyboard layout is not already loaded in keyboard input device **110**. Keyboard applet **114** locates the proper keyboard layout on a computer system coupled to network **108**. Keyboard layout **115** can be retrieved on demand from the network (step **308**).

Once downloaded, CPU **104** invokes a configuration method within keyboard applet **114** to configure keyboard input device **110** (step **310**). Keyboard applet **114** then stores keyboard layout information associated with the correct configuration of keyboard **110** in NVRAM **128** (step **312**).

Keyboard input device **110** displays symbols on each key corresponding to the keyboard layout (step **314**). Depending on the number of symbols and the complexity of symbols, various display technologies can be used to create the appropriate display key. For example, each key may include a keycap having a an LCD capable of displaying single byte or double byte characters. The display on each keycap may display a single symbol or multiple symbols depending on the application. When a user activates a key, the key provides computer system **101** with a corresponding symbol or symbols as displayed on the keycap.

Various techniques can be used to set the layout flag on the computer system. FIG. **4** is a flowchart illustrating exemplary steps used by keyboard applet **114** to select a layout flag. This flowchart relates to step **302** in FIG. **3**. Keyboard applet **124** initially checks for a layout flag as set by a smartcard device (step **402**). If a smartcard device is present, keyboard applet **114** sets the layout flag in computer system **101** to the value in the smartcard device (step **404**). For example, this enables a user to select a keyboard layout by inserting a smartcard device into the keyboard of the computer system. For an international traveler, a layout flag