Virtual Keyboard UI/Module design (v0.2)

02/26/2008 Naoyuki Ishimura

Preface

Virtual Keyboard is a GUI representation of keyboard on desktop screen. This document describes its requirements, features/UI and basic mechanism with diagrams including Input Method framework (iiim) module relationship.

Requirements

- It should enable to input characters without touching physical keyboard.
- It should enable to input characters which is not present on physical keyboard.
- It should be able to synchronize with current input mode (keyboard layout emulation by Input Method framework (iiim)).
- It should enable to create and add custom layout for current iiim keyboard emulation layout list

Features/UI

Virtual keyboard is the collection of push buttons basically, and each button represents one key of keyboard. The alignment sequence of keys and labels of each key can be selectable by user. The below list represents the features which virtual keyboard will provide.

- It has selection for keyboard geometry type such like 'Sun Type6 Unix keyboard', 'Generic 101' or 'Brazilian ABNT2'.
  - The base data for this will come from XKB geometry data.
  - Virtual keyboard shape and key sequence depends on this selection.
- It has selection for language type of keyboard such like 'Arabic', 'French' or 'Russian'.
  - The base data for this will come from XKB symbols data.
  - This selection determines the key top labels (which indicate expected characters when it is clicked).
  - The labels of key top reflect with modifier status. For example, if shift modifier is selected status, then labels correspond to shift status are appealing.

  See Appendix A: Visual effect with modifier keys

  The combination of geometry and language type determines virtual keyboard type.
- It has 'DESIGN' mode addition to normal input mode.
  - User can arrange the location of key, add new key or delete keys based on existing type of virtual keyboard and can save it for later use.
  - User can change the labels of each key to customize input characters.
  - User can assign more than one characters to one key for entering string with single action like single button click or single key press.
  - User can register the customized type as new IIIM keyboard layout emulation candidate.
  - Registering can be done per user or per system. Per system registration is allowed to user who has the appropriate permission.
  - User created custom virtual keyboard data is stored in separate xml file and other user or system can utilize it by coping it to $HOME/.iiim/vkb/types or /var/lib/iiim/vkb/types directory.
- It has option to toggle synchronization on/off with IIIM keyboard layout emulation.
  - When synchronization is on, then the virtual keyboard shows current physical keyboard

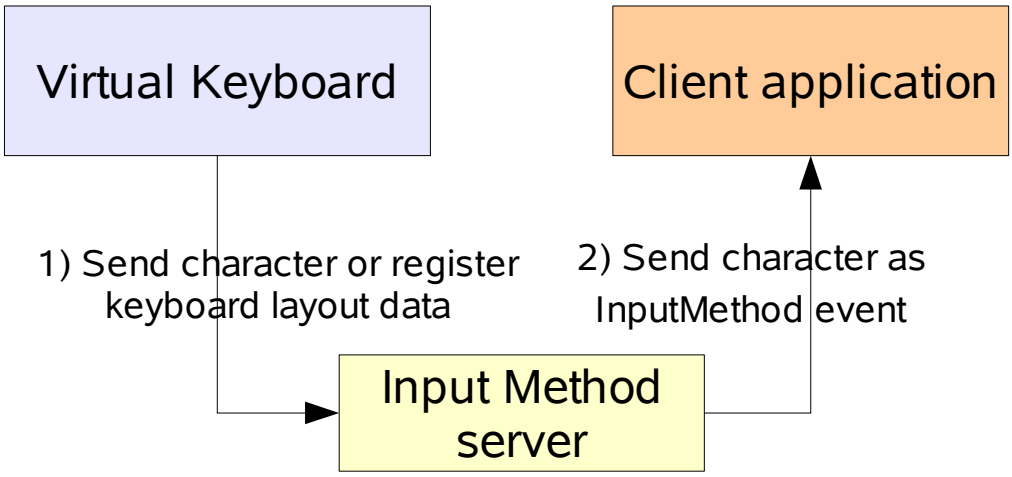
status. It means typing physical keyboard and virtual keyboard will produce the same character. User can confirm current emulated keyboard with virtual keyboard.
    - When synchronization is off, then virtual keyboard produces its own character independently with physical keyboard typing.
- It has three modes to display virtual keyboard.
    - Single frame shows one virtual keyboard and switching the type (geometry/language) changes the content of the frame. Multiple top level frames are supported.
    - Single top level frame shows multiple virtual keyboard as its internal frames.
    - Single top level frame has multiple tabs and each tab pane shows one virtual keyboard.
    - Its visual size can be scaled with frame resizing in each frame styles.

  See Appendix B - Multiple view with different styles
- It has clipboard which can be used pasting characters to other applications.
    - This enable inputting characters on virtual keyboard to Input Method unaware clients.
- It's invoked with menu (iiim-panel menu or JDS main menu – Application/Accessory), or executed by command 'iiim-virtual-keyboard'.

Module diagram

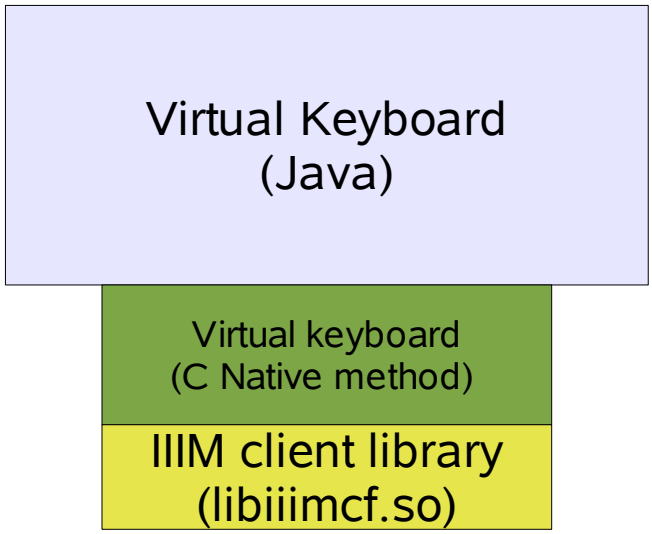
Basic

Virtual keyboard will send its generated event to Input Method server using iiim client library. And Input Method server will send that data to the latest focused Input Method client as Input Method event so that client application can handle it as normal Input Method commit event.



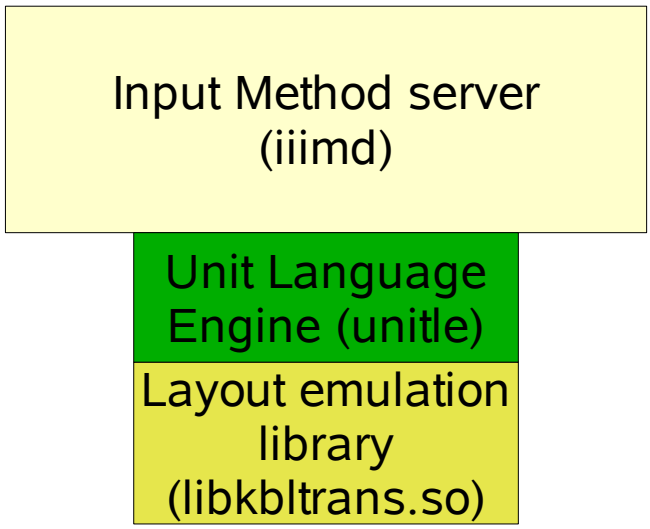
Virtual Keyboard

Virtual keyboard uses iiim client library called iiimcf.so which has the function to send event to Input Method server. And using the same library, it notifies to keyboard layout emulation engine that new layout data is available and should be activated it.



Input Method server

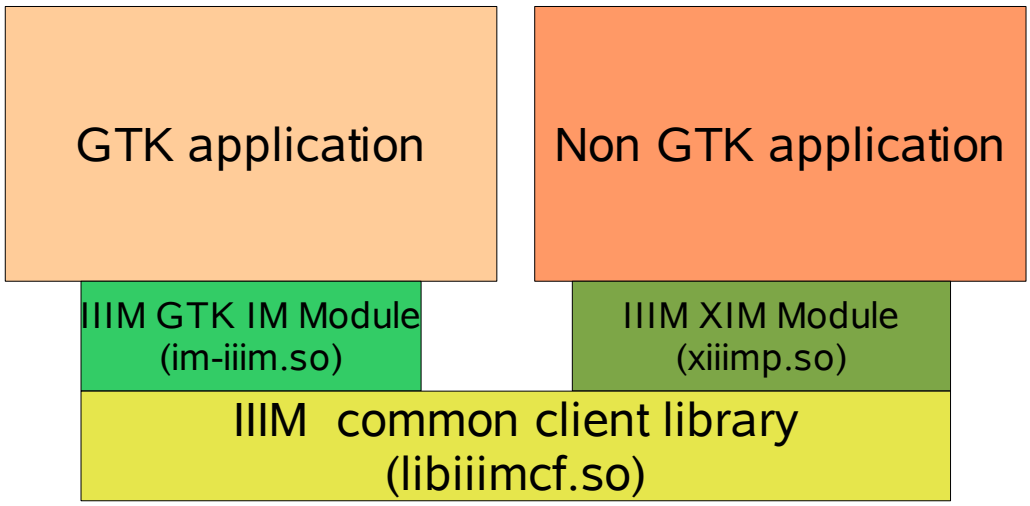
Input Method server (iiimd) receive event from virtual keyboard and send it to the last focused client window as commit event. And when it receives new layout data availability, it let unitle/libkbltrans to re-read layout data and add new data to its emulation list.



Client Application

The client application receive Input Method commit event to get characters from virtual keyboard. The application which uses Input Method (GTK IM or XIM) can get such event without any modification. The applications which do not use Input Method will not receive such event, so user

need to use copy/paste operation between virtual keyboard and application.



Appendix A: Visual effect with modifier keys

Shift modifier pressed

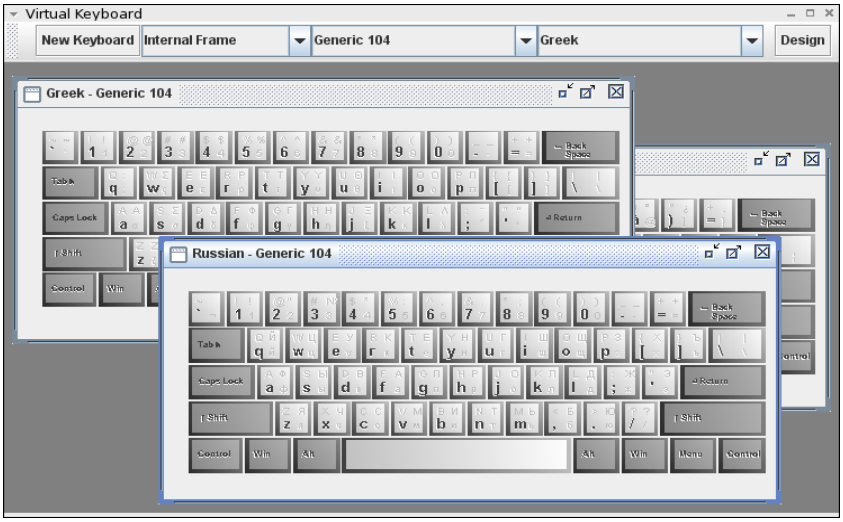
Caps modifier pressed

AltGraph + Shift modifier pressed
(Bulgarian)

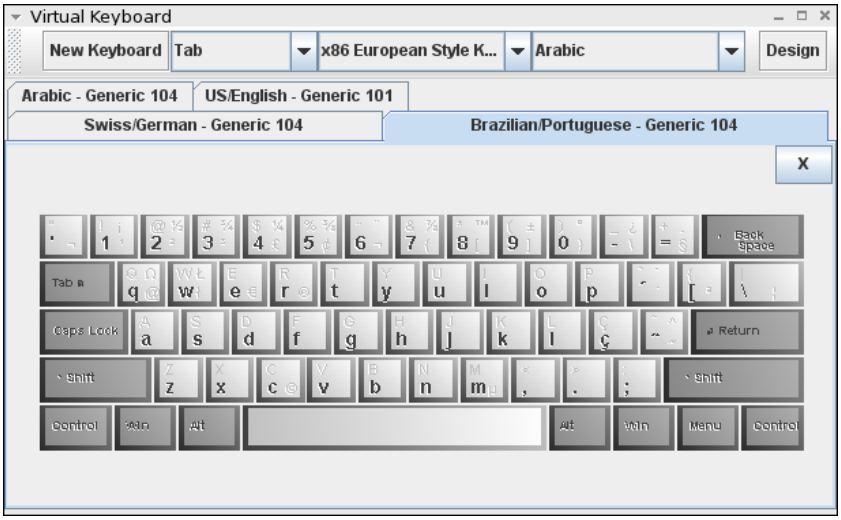
Appendix B: Multiple view with different styles

GUI controls are not designed yet. These screenshots are just for showing concept.

Multiple Frame style



Internal Frame style



Tab Frame style