

Input overview and plans



Daniel Stone
X Developer, OSSO
daniel.stone@nokia.com

Input overview

- Xi: input's redheaded stepchild
- Hotplug: fame and fortune through buzzwords
- MPX: more awesome than you think
- XKB: the badboy of X extensions
- DIX/DDX status
- Event refurbishment

Xi version zero point despair

- Developed for a very specific usecase — one mouse for normal UI interaction, plus a tablet to deal with graphics issues.
- But you don't want to use your tablet in the UI. That would be stupid.
- Xi thus completely ignored, because normal people just want to interact with their UI. GIMP basically the only app to make use of it.
- Exposes a device with an arbitrary number of buttons, axes (valuators, just to keep you on your toes), keys, et al.
- Scan the device list, and open your device -> exclusive event delivery.

Xi done right

- Embarrassingly, SGI actually did this right.
- Core event delivery is right: ask for button press/release, motion notify, etc.
- Xi event delivery ... is not.
- Useful model: core events, but also telling you which device is which.
- Gives you a list of devices (Logitech USB keyboard, Synaptics touchpad, Miscellaneous USB mouse), and lets you know when it changes.

Input hotplug

- Once all this is done, input hotplug becomes pretty trivial.
- 1100 lines for a full D-Bus and HAL implementation (either/or).
- Less than the amount needed for event generation and delivery.

MPX

- Multiple pointers, with multiple *cursors*.
- ... which means multiple focii.
- Multiple grabs mean head-exploding issues.
- Watch it on YouTube if you haven't seen it already.

XKB

- Exposes stupid binary format through the wire.
- API was defined in terms of implementation: unmitigated loss.
- New API that matches use patched on, but done as unintrusively (read, badly) as possible.
- Code is disastrous, enormous, slow, buggy.
- Yet, a very necessary evil.

XKB, redeemed

- Get rid of the old API, and make *everything* deal with RMLVO.
- You don't want KcCGST. Trust me.
- Fold xkbcomp into the server (volunteers?).
- Third attempt at writing RMLVO: first disk death, then laptop theft. (Backups are for other people.)
- Second attempt at writing xkbcomp-in-server.
- Would be fantastic to get in for 7.4.

DIX/DDX status

- Far too much was previously punted to the DDX.
- ... and duplicated among DDXes.
- ... and copied identically between DDXes.
- ... and borrowed from another DDX, yet slightly tweaked.
- (laughter, applause)
- input-hotplug's primary aim was to move as much to the DIX as possible.
- DDX now just feeds in a couple of events, but is still far too involved.

Event refurbishment

- Get(Pointer,Keyboard,...)Events return a bunch of xEvents.
- DDX iterates over events, submitting them to the DIX.
- During event processing, DIX attempts to infer from the events: trying to recreate information.
- Instead of passing around xEvents, just pass around a new struct which tells you everything you need.
- Deal with a single struct describing a single event (not Event).
- During input processing, explode into multiple xEvents when absolutely necessary.

Remaining issues

- High keycodes — just can't solve this without a protocol bump.
- More than four layouts simultaneously — either a disastrous server hack, or a protocol bump.
- Multiple pointers — if only someone could solve this intractable problem!
- Code quality — XKB is measurably the worst code in X, Xi suffers from even more shame than it's due.
- Bugs — 1.4 has a couple of embarrassing bugs, will be fixed in 1.4.1.
- input-hotplug adoption — zero.

Questions?

NOKIA
Connecting People

Thanks.

NOKIA
Connecting People