XML: An Xlib Compatibility Layer for XCB

http://xcb.cs.pdx.edu/

Jamey Sharp
Computer Science Department
Portland State University
Portland, Oregon USA 97207-0771

Bart Massey

{jamey,sharp}@cs.pdx.edu
Overview

Summary •

XCL •

XCB •

The X Window System •
Thesis

Application of modern tools and X experience to X Window System client libraries can produce modern code

- by separation of concerns
- by design of new interfaces
- by re-implementing existing interfaces
X Window System Architecture

What is X?
X Strengths

- Client and server implementations are interoperable
- Client libraries for many languages and platforms
- Servers from XFree86, Sun, others
- Network protocol specification separates hardware from apps
- \[ \text{toolskits} \]
- \[ \text{window managers} \]
- \[ \text{separates policy from mechanism} \]
- Well designed

**X**
Xlib's role in the X Window System

**X Server**

**X protocol**

**Xlib**

**Toolkit**

**Client**

**Xlib: C Protocol Binding**
Motivations for XCB

Why change the Xlib API?

- Smaller and simpler API and implementation
- New API enables a light-weight, flexible implementation style.
- Some optimizations affect API requirements
- Latency is a problem for existing X applications
- XCB API supports latency hiding in all cases
- Xlib API supports latency hiding in special cases
- XCB API supports latency hiding in all cases
- Latency is a problem for existing X applications
- Some optimizations affect API requirements
- Smaller and simpler API and implementation

Support both single and multi-threaded clients with one API

(c.f. Mason, "X meets Z")
XCBEvaluation

ApI

concept has been proven: modern implementation with new

- code size is particularly good: 27kB
- expected benefits have been successfully achieved
- most planned features implemented

XCBEvaluation
XCL's role compared with Xlib's

XCL: an Xlib Compatibility Layer for XCB
Motivations for XCL

Why re-invent Xlib?

- Significant history: > 15 years worth of software uses Xlib API
- In small environments like hand-held computers, Xlib implementation is big
- XCB implementation is small, but API is incompatible with Xlib applications
- Aid transition to XCB through XCL
- Anticipate possible benefits in performance, latency, reliability
New design and implementation enables many of XCB's benefits without changing Xlib API.

- More uniform optimization
- Easier maintenance
- Easier request marshaling in not just XDrawPoint but also XDrawPoints

XCB Strengths
Some applications tested without error, e.g.

- rxvt: perfect behavior
- gw: nearly perfect

XCL+ECL is 55KB; Xlib is about 66KB

- Concept has been proven: modern implementation of Xlib API
- XCL performance comparable to Xlib
- XCL+ECL is 55KB; Xlib is about 66KB
- Some applications tested without error, e.g.

XCL Evaluation
Replace core font rendering with Xft

Implement caches on XCB

Validate against existing Xlib-based toolkits (Qt, GTK+)

Analyze: other Xlib functionality ← additional modules

Complete XCL (extension support)

Future Work
Related Work

- Standalone client: AppleLogic, and Jacobson’s 1991 IOCCC
- Conway’s Game of Life runs on X desktop
- Libraries for other languages:
  - ML: eXene
  - Smalltalk: STIX
  - Common Lisp: CLX
  - Java client libraries: XTG, Escher

- Libraries for other languages:
The work described in this talk is a joint project with PSU Prof. Bart Massey, XFree86 Core Team member Keith Packard, and high-school student Andy Howe.

We are grateful for the significant contributions of Jim Gettys, one of the original authors of Xlib and of the design of the X Window System.

We are grateful for the significant contributions of Jim Gettys, one of the original authors of Xlib and of the design of the X Window System.

And we are greatly indebted to the Computer Science department of Portland State University for supporting this research and enabling us to present the work at the Usenix Annual Technical Conference 2002, and at the XFree86 Technical Conference 2001.

Acknowledgments
Current implementations of XCL and XCB are freely available under an MIT-style license at http://xcb.cs.pdx.edu/.