Software Maintenance in a Campus Environment:

The Xhier Approach

John Sellens
Math Faculty Computing Facilty
and
Department of Computing Services
University of Waterloo
jmsellens@UWaterloo.ca

Quick Summary of Xhier

- Xhier is a system for software installation and distribution
- Runs on many different machines
- Highly automated
- Used extensively at the University of Waterloo
  $>$300 machines, $>$350 software packages,
  $>$1,600 commands, $>$21,000 package installations

Purpose of the Paper

- Describe one approach to this problem that seems to work well for us
- Not necessarily the best answer for everyone
- Some of the things that we have learned can be applied by others to their situations

History and Motivation

- One DEC VAX 11/780 with 4.2BSD
- Easy to modify, complete source
- Extends to more than one machine with `rcp` and `rdist`
- New Sun3, Sun4, Sequent, MIPS, Ultrix machines, more frequent OS updates, make things more complicated.

The Problem to be Solved

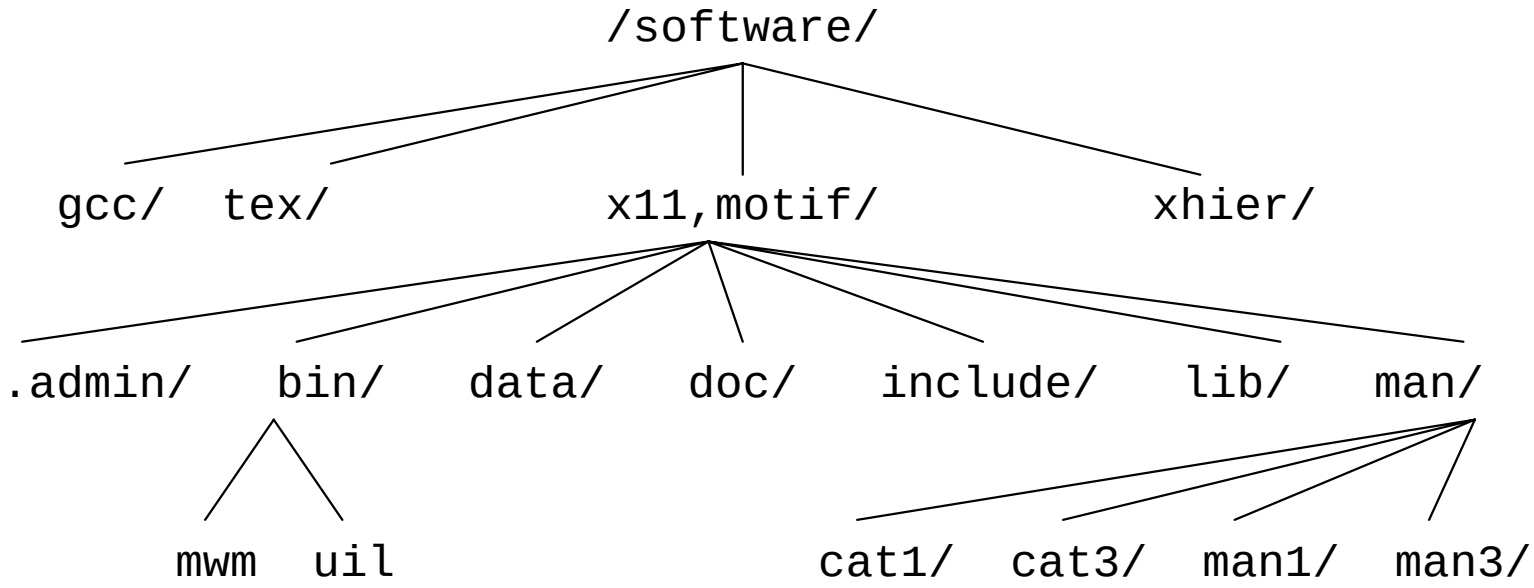
- MFCF had to “support” many machines, many administrators, many different needs
- Lots of useful software available
- Need some way to install and distribute it
- Must be flexible, configurable, and automatic

Basic Approach of Xhier

- Want to minimize changes to vendor systems
- Put our software in a separate hierarchy, under `/software`
- Group software into **packages** of related software
- Define a standard structure for packages to follow
- Provide *lots* of tools for dealing with packages

Package Structure

- Each package directory contains everything to do with that package



Package Specific Support Files

- `.admin` directory
  - Files that govern installation and distribution
  - `Install Dependencies Maintainer Options ...`
- `export` directory
  - Files for use outside the package
  - `boottime crontab services inetd.conf passwd group ...`

Complicating Everything

- Separate file hierarchies based on file type are useful
  - No duplication of shared files on heterogeneous servers
    — similar to Sun's `/usr/share` directory
  - Simplifies remote mounts — mount only the parts you need
  - Nice to be able to easily put things in different partitions
    e.g. "spool" files

- `/software` is organized by file type under `/.software`
  — `share arch spool admin regional local`

Important Tools

- `xh-install` — runs `Install`, modifies system files, makes package available for use
- `xh-distribute` — sends a package to other machines
- `xh-sdist` — sends a package's source to other machines and runs `make`; makes doing updates very easy
- `xh-maintenance` — run weekly, does distribution, installation, error and sanity checking

User Accessibility

- Want the system to be as transparent as possible for users
- Use search rules as much as possible
- `showpath` command used to set `PATH` and `MANPATH`
- `xh-make-links` links package files into system directories `/usr/include /usr/local/lib`
- Also makes link directory of all commands, so only one special directory has to be added to `PATH`

Important Things We Have Learned

- Automation of almost everything is important
  – automation means *Be Careful*
- Hard to organize NFS mounts of parts of `/software`
- System administrators must understand the system
  – “public relations” is important
- Not as easy or as small as we had hoped it would be
- Lots of subtle and obscure things

Conclusions

- It works
- It's possible to support lots and lots of machines, with minimal per-machine effort
- It's easy to install and distribute new software and bug fixes, even across multiple architectures


THE