Custom Debian Distributions

Sense, scope, techniques and future

Andreas Tille

Debian Conference 5

Helsinki, 11. July, 2005

Structure

1. Past
    - Short history
    - Obliged to users
    - Limits to Growth
    - Deriving from Debian
2. Present
    - Grouping of packages
    - Adapting package configuration
    - Documentation
3. Future
    - Technical
    - Structural

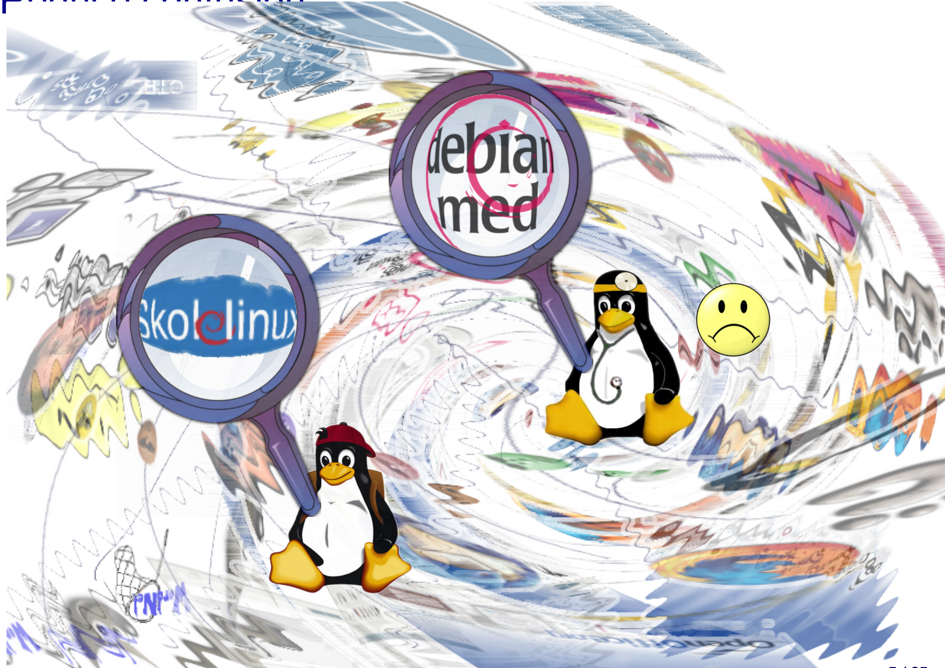
Origin

- Debian internal projects
  - Debian-Junior (2000)
  - Debian-Med (2002)
  - Debian-Edu (2002)
- Amongst technical internal projects
- Just grouping packages in meta packages

Evolution

- DebConf 3 renaming: *Custom Debian Distribution*
  - Distinction between technical projects and CDDs
  - Start of technical cooperation
- Forks coming back
  - SkoleLinux became Debian-Edu
  - DeMuDi

Reduce confusion



Customising Debian

- Debian $>$ 10000 packages
- Users interested in *subset*
- Groups of specialised users
- Easy installation and configuration
- **No fork** from Debian

Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead

Particular features of Debian

- Maintainer: Missing link between developers and users
- Principles of Free Software at distribution level: Independence of commercial interest of producer

Why including special applications

- Debian becomes larger and larger

⟶ How many packages are good for Debian?

- Bring special applications under quality control
- Attract wider user base

⟶ How many packages are good for our users?

Do we scale well?

1. Number of involved people
2. Number of packages
3. Number of architectures
4. Number of bugs
5. Number of users
6. Number of derivatives
7. Time span between releases
8. Number of attendees of Debconf
9. Amount of money sponsored for Debconf

Sub-structuring Debian

- Change of quantity into quality (Hegel)
- To determine at the right moment the critical point where quantity changes into quality is one of the most important and difficult tasks in all the spheres of knowledge (Trotzki)
- Evolution . . . through quantitative transformations passing into qualitative (Darwin principle)

⟶ CDDs can deal with nearly all dimensions of growth inside Debian

CDD scales well

Problem		CDD-solution
No. people	Smaller projects	Focussing work better control
No. packages	Subset	Closed package set
No. bugs	CDD relevant	Stronger focus
No. users	Specialists	Interested users known needs
No. derivatives	Customisation	Keep inside
Time span	Releases	Independent

Reasons

Commercial reasons

- Add something to Debian
  - Add non-free software
  - Add own installer, desktop configuration, etc.
- Service company needs a brand

Political reasons

- On behalf of government or organisation
- Granted project

Misconceptions about Debian

- Impossible to work together with Debian
- Debian is an inflexible beast
- Ignore possibilities of DoOcracy

Meta packages contain

- Relations to other Debian packages

  Depends, recommends, suggests
  Conflicts ?

- User menus

  Maintained via role based tools

- Configuration

User roles

- Not in the menu == not available
- Reduce menu to interesting part

There are CDD users outside

Thankful user

```
Date:  Tue, 5 Jul 2005
From:  Djoume SALVETTI <salvetti@crans.org>
To:  Andreas Tille <tille@debian.org>
Subject:  Re:  cdd ACCEPTED

Oups!  Sorry for the noise, I have misconfigured my debarchiver.  :-/

(BTW, many thanks for cdd-dev, I foudn it very usefull!  :-)

Djoume
```

Explanation

```
I dont really build a CDD. I'm working in a enginnering school (IFITEP) and I'm
switching all PC (a hundred) from Redhat to Debian sarge.
Each courses require a set of software, so I'm building a meta-package for each
courses (debian-ifitep-java, debian-ifitep-maths, etc..)  and adjust the dependency
according to the need of the teacher.
I will certainly have to adjust the dependency during the school year (for example if
a teacher have forgotten to tell me something) and the best way I have found to keep
the PC in sync is to :
- make the meta package as explained before
- make a local repository
- auto-upgrade the PC at boot time and with a cron
With the help of your CDD-HOWTO, I have found it very easy to build meta-package and
I'have also decided to make a "private" meta-package with a selection of packages I
can't live without and use it when I made a new debian install.
```

Debconf

- Relay on package maintainer
- Cope with overworked / lazy / stubborn maintainers
- Group maintenance

Cfengine

- Fall-back if `debconf` not possible
- Give local admin helping hand
- Problems on upgrades might be solved by `cddtk`
    - Wrapper around apt
    - Store original config files
    - Move these in place before upgrades
    - Redo CDD specific configuration

Obliged to end users

- *CDD documentation*
- Web-tools
  - Currently nothing
  - Possibility to do basic things automatically
- Very different between CDDs

CDD toolkit

- *New generation toolkit*
  - Replace *cdd-dev* to build meta packages
  - Alternatively resolve dependencies without meta packages
  - Techniques for simple configuration

Live-CDs and installers

- Live CD – nice to have
- Perfect to show off
- For *real* work: installer
- Make it “user-just-pressing-enter” like

Documentation

- Better web presence
  - Automatically graphing dependency map
  - Unified web pages
  - Enhancing visibility
- Bug Tracking System
  - Tag WNPP bugs
  - Idea: View on BTS for CDD dependencies
  - Enhanced CDD-centric BTS user interface
- User documentation

Future of Debian

- CDD solve structural problems inside Debian
- Fit user interests
- Makes Debian stronger if done the right way

Also known as: The last, final step towards Total World Domination!

This talk can be found at
http://people.debian.org/˜tille/talks/
Andreas Tille <tille@debian.org>