

# OpenEmbedded

If we had to start all over, what would we do different?

ELC-E 2015 - Dublin  
Koen Kooi <[koen.kooi@linaro.org](mailto:koen.kooi@linaro.org)>

# Vanity slide

Hi, my name is...

# What's in this presentation

- Historical overview
- Current situation
- My grand vision of OE
- A lot of room for questions

# Don't hesitate to interrupt me!

Seriously, if you have a question or need some clarification don't hesitate to interrupt.

If I say something that's blatantly untrue, don't hesitate to shout.

# Timeline

OE - founded late 2003

Yocto Project announcement - ELC-E Cambridge 2010

OE-core - started 2010

# A bit of historic background

OE was started to solve the 'one source tree, multiple boards' use case, which buildroot, at the time, couldn't do. It was ocused on *configuration management*.

Chris Larson needed that for both work and hobby purposes. His hobby project was OpenZaurus which explains a lot of the quirks in early OE.

The next slides will have bullet points again.

# OpenEmbedded design principles

Orthogonal:

- MACHINE.conf
- DISTRO.conf
- image.bb

So you can build any image for any machine for any distro.

# OpenEmbedded design principles

Orthogonal:

- MACHINE.conf
- DISTRO.conf
- image.bb

So you can build any image for any machine for any distro. In theory.

# MACHINE.conf

- Describes the hardware:
  - selects tuning for cpu arch e.g. cortex-a53
  - selects kernel recipe e.g. linux-linaro-aarch64
  - sets MACHINE\_FEATURES e.g. “touchscreen mmc usb wifi bt”
  - optionally selects bootloader e.g. edk2
  - optionally selects extra kernel modules e.g. kernel-module-gpu-driver
  - optionally selects extra packages e.g. gpu-firmware-package
  - optionally sets storage quirks like ubifs/jffs parameters
- It must **NOT** meddle in DISTRO affairs like:
  - floating point format
  - software versions outside of kernel, bootloader

# DISTRO.conf

- Set policy
  - ABI
  - compiler
  - C library
  - init system
  - package versions
  - package manager
  - PACKAGE\_CONFIG tweaks
- Can do whatever it wants

# image.bb

- Selects packages to install
- Can add custom processing steps
  - write timestamp
  - make it safe for RO usage
  - include initramfs
  - generate manifest
- Thanks to scope can't really influence anything outside of the image

# OpenEmbedded design principles

This was violated pretty much from the start:

- opensimpad-image only builds for DISTRO=opensimpad, MACHINE=simpad
- MACHINE=nslu2(le/be) only builds for openslug/slugos.
- OpenZaurus kernels relied on the compiler to be bug compatible with gcc 2.95.3

Biggest distro offenders: Familiar, OpenZaurus and their clones.

# Layers

- Made popular by the Yocto Project initiative in 2010
- No real rules
  - “Yocto Project Compatible” stamp of approval
- Silo mentality

# Layers

Show <http://layers.openembedded.org> in a browser

# Contribution guidelines

- Basically free for all
- Review was met with violent opposition
- Switched to pull model after OE-core was created.

My grand vision of OE

# Document use cases

- SDK
  - Relocatable
  - Updatable
- board/cpu/ISA bringup
  - site config files
  - common pain points (openssl configure)
- product
  - tight combo of MACHINE/DISTRO/image.bb

# Be explicit in what it's not

- Not for application developers
- Not a distro
- Not yocto
- Not for everyone

# Respect the end user/product

- External binary toolchains
- GPU binaries
- Ancient kernels

# Compete in challenges with other buildsystems

- Have challenges at popular conferences (FOSDEM, ELC)
  - “Websserver on beaglebone in 16MiB or less”
  - “Document #exactsteps needed to blink LED on C.H.I.P.”
  - “Build UEFI helloworld”

# Metrics for every step

- time spent in tasks
- disk space needed in `#{WORKDIR}`
- files accessed
- buildhistory
- licenses
- check rpath/shlibs/etc

# Decouple rootfs from build

- Only generate a tarball + directory with packages used in said tarball
- Have external tools to generate ext4/ISO/fastboot/UBI
- Have BSPs provide tools/plugins

# Better consistency

- linux-libc-headers vs OLDEST\_KERNEL
- \*\_FEATURES
- Better scoping
  - no poking at IMAGE\_INSTALL outside of image.bb
  - no poking at ABI from MACHINE.conf
- Better tracing

# Empower maintainers

- Kernel style checkpatch
- Kernel style get-maintainer
- Maintainers have veto

# No more Yocto syllogisms

1. This needs to be changed
2. This patch is a change
3. Ergo: This patch is needed



# Abstract

The OpenEmbedded project has entered its second decade now and can be called fairly successful. With the 10+ years of experience and user feedback we have now, what would we do differently? This presentation will talk about how components are split now like bitbake and metadata being separate projects as well as how they are governed. A large portion of this presentation will be about explaining the fundamental ideas behind OE (machine, distro and image are orthogonal) and how it has influenced the design and workflow and how we would do it differently with the experience we have now. The last portion will deal with commercial interests like companies using OE for shipping products or for testing and validating their software like Linaro does.