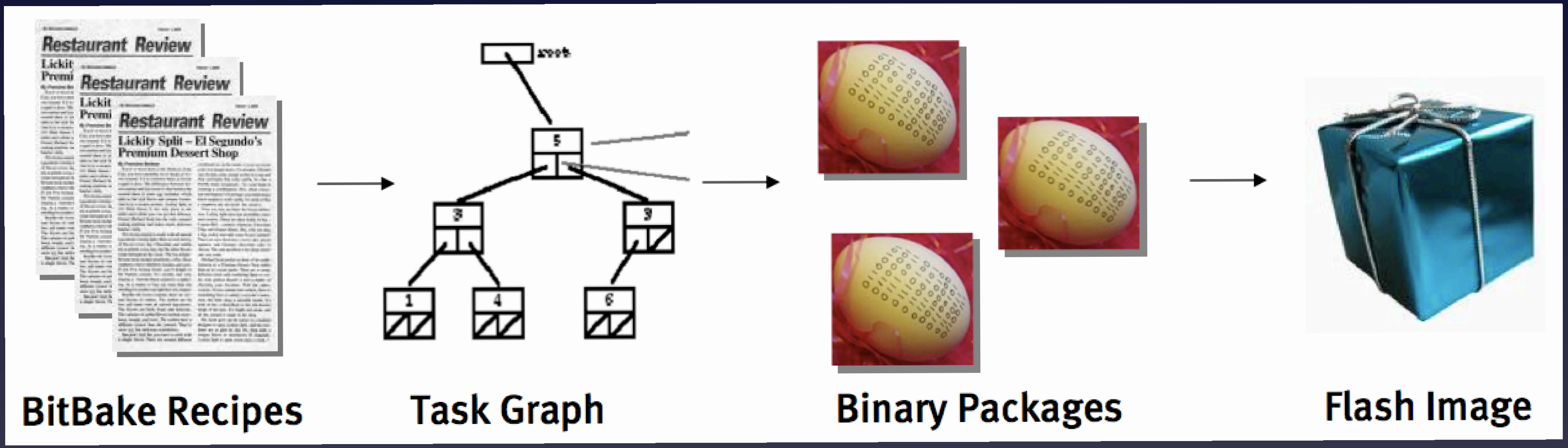
OpenEmbedded

Easy QA, repeatability and retargetting

Koen Kooi
Michael ‘Mickey’ Lauer
Richard ‘RP’ Purdie
Holger ‘zecke’ Freyther

The BitBake Task Executor



```
# bitbake foo
```

- parsing data from all recipes it’s instructed to find
- For each recipe
  - Builds a storage area to hold the metadata that comes from
    the local environment, the recipe itself, the data in the build classes a recipe include.
  - Computes task dependencies
- Builds a combined task graph containing all tasks from all recipes
- Builds all task dependencies for `foo`
  - generates a shell script on-the-fly out of the metadata
  - runs the shell script
- Builds all tasks listed in recipe providing `foo`

The OpenEmbedded metadata repository

A repository containing **everything** you need to build software from scratch

- Buildclasses containing common tasks, e.g. support forbuildsystems
  - autotools
  - scons
  - qmake
  - gettext
  - python distutils
- Machine configurations for
  - common architectures, developer boards
  - PDAs like the Sharp Zaurus, HP iPAQs and various HTC devices
  - Internet tablets like the Siemens Simpad, Nokia 770
  - Routers like the Linksys wrt54g
  - NAS devices like the Linksys NSLU2 and Thecus n2100
  - GSM phones like the Motorola A780, FIC Neo1973, HTC Universal
- Distribution policies like packaging, naming, preferred toolchains, version, ...

- Over 4000 recipes describing how to build software
  - Description and licenses
  - Source location and patches
  - Dependencies

Quality Assurance

- Check for dangling symlinks
- Check architecture
- Check for stray .la and .so files
- Check for stray debug symbols

Repeatability

- Lock down program releases via

```
PREFERRED_VERSION_<foo> = 20070402
```

- Lock down program snapshots via

```
SRCDATE_<bar> = 20070402
```

- cache tarballs on a central server

Retargetting

Building different things for different targets

Set up the basics

- Download bitbake and OE according to the GettingStarted document
- Point BBPATH to your bitbake checkout
- Point BBFILES to your OE checkout
- Set DL_DIR and TMPDIR if you want

Retargetting

Building different things for different targets

build for a ARM smartphone

```
echo “MACHINE=a780”          >  conf/auto.conf
echo “DISTRO=angstrom-2007.1” >> conf/auto.conf
bitbake smartphone-image
```

Retargetting

Building different things for different targets

build for an OMAP devboard

```
echo “MACHINE=omap5912osk”    >  conf/auto.conf
echo “DISTRO=angstrom-2007.1” >> conf/auto.conf
bitbake dspgateway-image
```

Retargetting

Building different things for different targets

build uclibc for a PowerPC board

```
echo “MACHINE=efika”              >  conf/auto.conf
echo “DISTRO=angstrom-2007.1” >> conf/auto.conf
echo “ANGSTROM_MODE=uclibc”   >> conf/auto.conf
bitbake console-image
```

Retargetting

Building different things for different targets

build for a strongARM tablet

```
echo “MACHINE=simpad”                  >  conf/auto.conf
echo “DISTRO=angstrom-2007.1-oabi” >> conf/auto.conf
bitbake tablet-image
```

Ångström example #1

Provide source mirrors:

classes/angstrom-mirrors.bbclass:

```
MIRRORS_append () {
ftp://.*/.*/    http://www.angstrom-distribution.org/unstable/sources/
http://.*/.*/   http://www.angstrom-distribution.org/unstable/sources/
}
```

Ångström example #2

Make the c library selectable:

conf/distro/include/angstrom.inc

```
# Can be "glibc" and "uclibc"
ANGSTROM_MODE ?= "glibc"

DEPLOY_DIR = "${TMPDIR}/deploy/${ANGSTROM_MODE}"
require conf/distro/include/angstrom-${ANGSTROM_MODE}.inc
```

conf/distro/include/angstrom-glibc.inc

```
PREFERRED_PROVIDER_virtual/libiconv ?= "glibc"
PREFERRED_PROVIDER_virtual/libintl ?= "glibc"
PREFERRED_PROVIDER_virtual/libc ?= "glibc"

TARGET_OS = "linux${@['','-gnueabi'][bb.data.getVar('TARGET_ARCH',d,1) in ['arm', 'armeb']]}"
```

conf/distro/include/angstrom-uclibc.inc

```
PREFERRED_PROVIDER_virtual/libc = "uclibc"
PREFERRED_PROVIDER_virtual/libiconv ?= "libiconv"
PREFERRED_PROVIDER_virtual/libintl ?= "gettext"

USE_NLS ?= "no"

TARGET_OS = "linux${@['-uclibc','-uclibcgnueabi'][bb.data.getVar('TARGET_ARCH',d,1) in ['arm', 'armeb']]}"
```

Ångström example #3

Brand it!

conf/distro/include/angstrom.inc

```
TARGET_VENDOR = "-angstrom"
```

$ ls tmp/cross/bin | grep gcc

```
arm-angstrom-linux-gcc
arm-angstrom-linux-gnueabi-gcc
i686-angstrom-linux-gcc
powerpc-angstrom-linux-gcc
```

Recent Changes

- Threaded bitbake
    - saves buildtime even on UP boxes
- Debian package format support
- Scratchbox SDK generation
- Qemu-image generation

Changes to come

- LLVM integration
- Better QA
  - Automated regression testing
  - Stable snapshots
- Even more qemu integration
- Public buildserver
  - once remote bitbake has been implemented

The bigger picture

Creating a community of specialists

To solve problems, a build-system is not enough, so we have:

- Kernel hackers
- Toolchain hackers
- Application hackers
- Framework architects
- System integrators

Hire us!

- A few companies offer OE consulting
- We have freelancers as well
- Or invite us to conferences