17-708 SOFTWARE PRODUCT LINES: CONCEPTS AND IMPLEMENTATION

THE LINUX KERNEL – CASE STUDY

CHRISTIAN KAESTNER
CARNEGIE MELLON UNIVERSITY
INSTITUTE FOR SOFTWARE RESEARCH

READING ASSIGNMENT OCT 28

Apel, S., Batory, D., Kästner, C., & Saake, G. (2013). Feature-Oriented Software Product Lines. Berlin: Springer.

Chapter 3, 6.1

Apel, Sven, Christian Kastner, and Christian Lengauer. "Language-independent and automated software composition: The FeatureHouse experience." Software Engineering, IEEE Transactions on 39, no. 1 (2013): 63-79.

Sect 1-3

LEARNING GOALS

Understand how the Linux kernel developers implement variability

Understand common practices and guidelines

VARIABILITY IN THE LINUX KERNEL

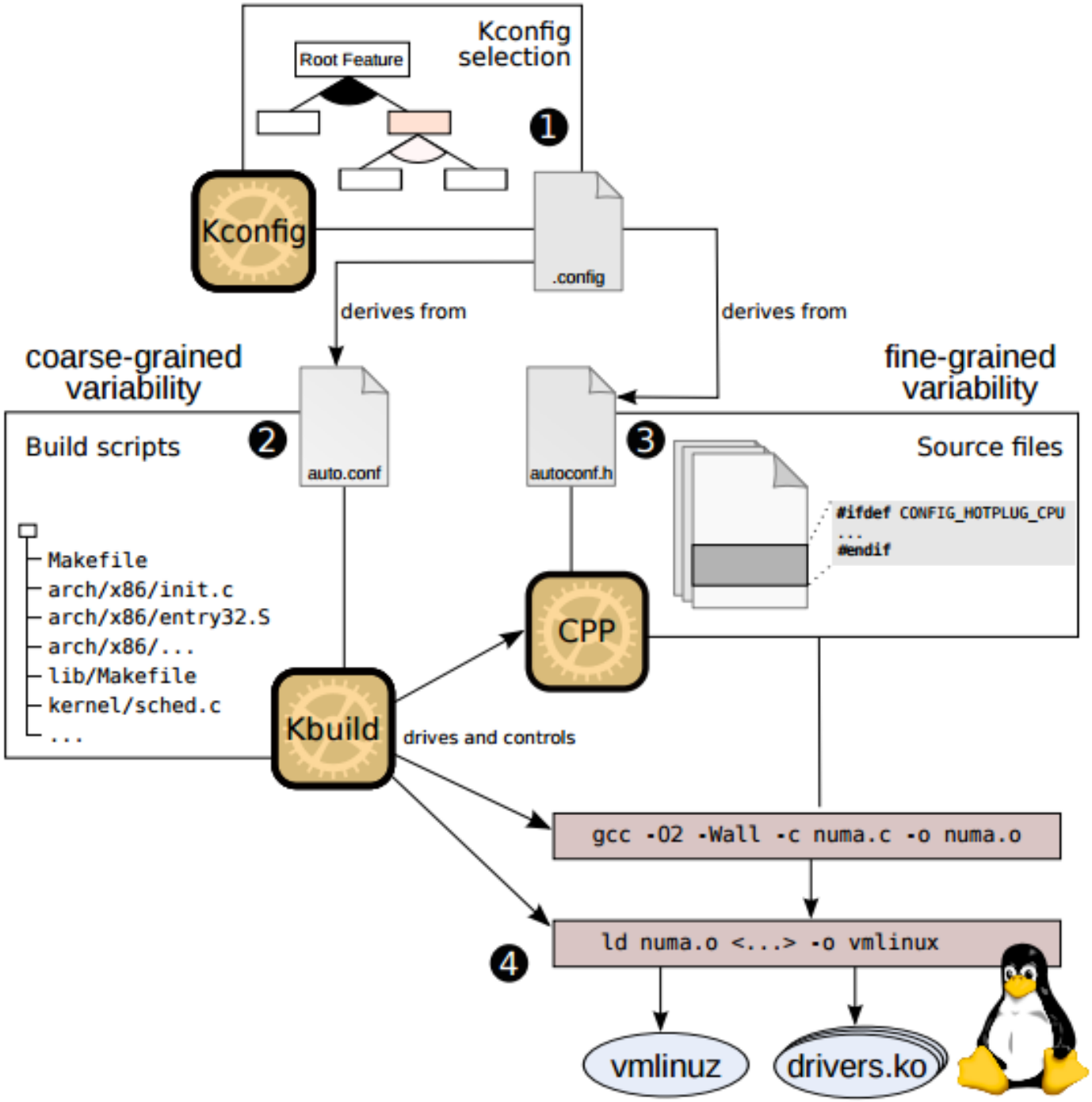
Preprocessor

Build system (kbuild)

Module system

Runtime variability

Variability model (kconfig)



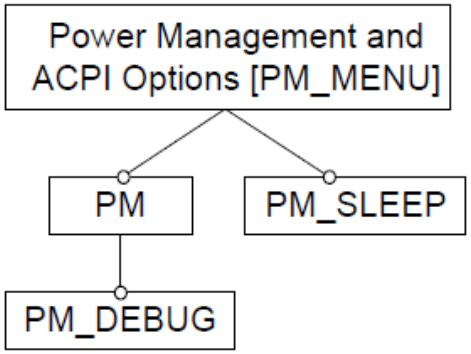
Dietrich, Christian, Reinhard Tartler, Wolfgang Schröder-Preikshat, and Daniel Lohmann. "Understanding Linux feature distribution." In *Proceedings of the 2012 workshop on Modularity in Systems Software*, pp. 15-20. ACM, 2012.

```
menu "Power management and ACPI options"
  depends on !X86_VOYAGER

  config PM
    bool "Power Management support"
    depends on !IA64_HP_SIM

  config PM_DEBUG
    bool "Power Management Debug Support"
    depends on PM

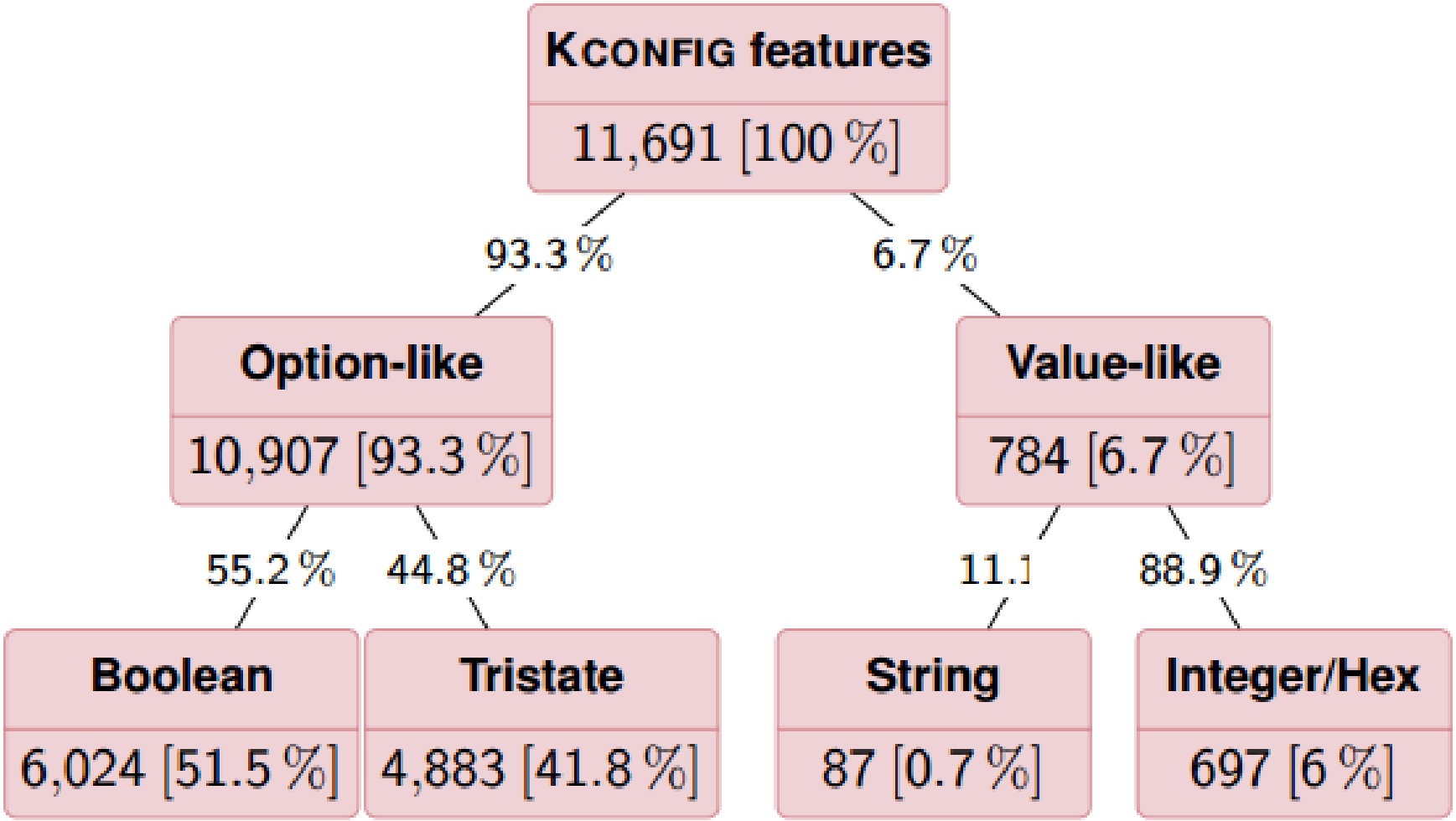
  config PM_SLEEP
    bool
    depends on SUSPEND || HIBERNATION
                       || XEN_SAVE_RESTORE
    default y
endmenu
```



$$PM_MENU \rightarrow \neg\ X86_VOYAGER$$
$$PM \rightarrow \neg\ IA64_HP_SIM$$
$$PM_SLEEP \leftrightarrow SUSPEND \mid\mid HIBERNATION \mid\mid XEN_SAVE_RESTORE$$

Fig. 1. A simple Kconfig model (left) and the corresponding feature model (right)

Lotufo, R., She, S., Berger, T., Czarnecki, K., & Wąsowski, A. (2010). Evolution of the linux kernel variability model. In *Software Product Lines: Going Beyond* (pp. 136-150). Springer Berlin Heidelberg.



Dietrich, Christian, Reinhard Tartler, Wolfgang Schröder-Preikshat, and Daniel Lohmann. "Understanding Linux feature distribution." In *Proceedings of the 2012 workshop on Modularity in Systems Software*, pp. 15-20. ACM, 2012.

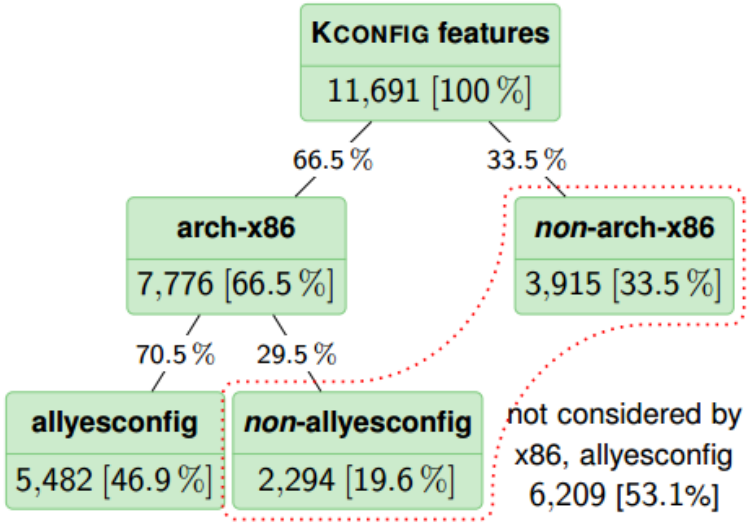


Figure 4. Coverarge of arch-x86 / allyesconfig

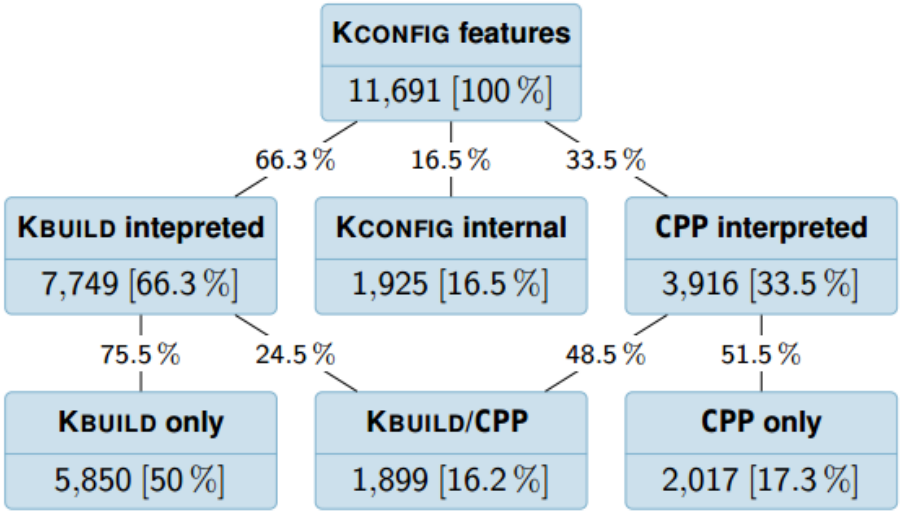
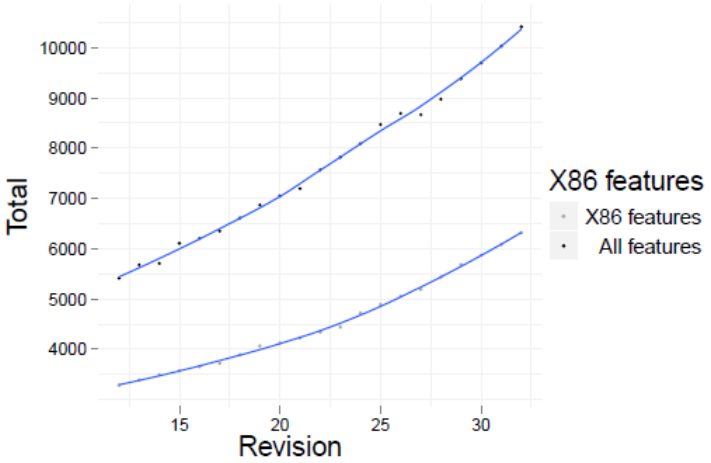


Figure 5. Distribution of features by granularity

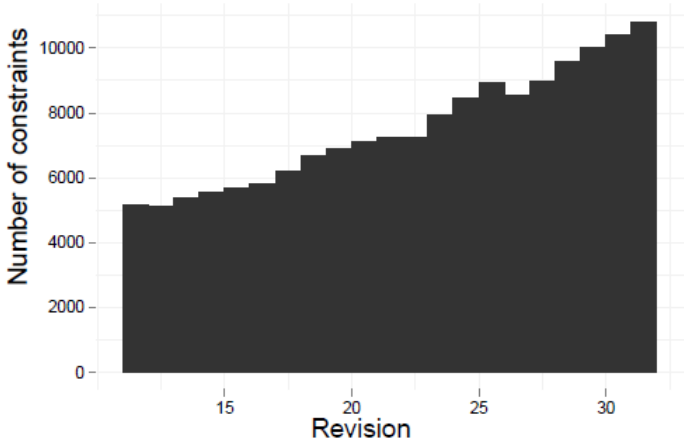
Dietrich, Christian, Reinhard Tartler, Wolfgang Schröder-Preikshat, and Daniel Lohmann. "Understanding Linux feature distribution." In *Proceedings of the 2012 workshop on Modularity in Systems Software*, pp. 15-20. ACM, 2012.

KBUILD

[demo]



(a) Total number of features in Linux Kconfig and x86 architecture.



(b) # of constraints per revision

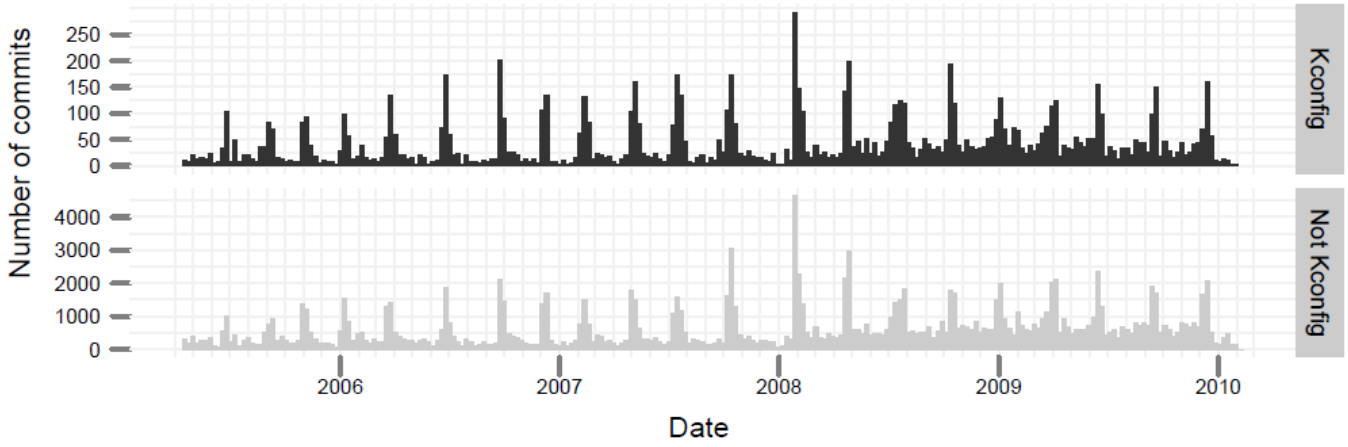


Fig. 3. Number of commits per week that touch Kconfig files compared to number of commits that do not. Each spike matches one of the 21 revisions analyzed.

Lotufo, R., She, S., Berger, T., Czarnecki, K., & Wąsowski, A. (2010). Evolution of the linux kernel variability model. In *Software Product Lines: Going Beyond* (pp. 136-150). Springer Berlin Heidelberg.

5. FINDINGS

We summarize our findings as follows. (i) Most features in the Linux kernel are modular and cause little scattering when added. Hence, we conjecture that the practice of keeping high modularity and low scattering mitigates the negative effects of ifdef-based variation points [24, 7, 6], allowing the kernel to continuously evolve. Furthermore, as the

Passos, L., Guo, J., Teixeira, L., Czarnecki, K., Wąsowski, A., & Borba, P. (2013, August). Coevolution of variability models and related artifacts: a case study from the Linux kernel. In *Proceedings of the 17th International Software Product Line Conference* (pp. 91-100). ACM.

drivers/staging/comedi/Kconfig

```
menuconfig COMEDI_PCI_DRIVERS
   tristate "Comedi PCI drivers"
...
+config COMEDI_ME4000
+   tristate "Meilhaus ME-4000 support"
+   help
+     Enable support for Meilhaus PCI data acquisition cards
+     ME-4650, ME-4670i, ME-4680, ME-4680i and ME-4680
...
```

drivers/staging/comedi/drivers/Makefile

```
-obj-$(CONFIG_COMEDI_PCI_DRIVERS) += me4000.o
+obj-$(CONFIG_COMEDI_ME4000) += me4000.o
```

drivers/media/video/Kconfig

```
config DISPLAY_DAVINCI_DM646X_EVM
   help
-     Support for DaVinci based display device.
+     Support for DM6467 based display device.

+config CAPTURE_DAVINCI_DM646X_EVM
+   tristate "DM646x EVM Video Capture"
+   depends on VIDEO_DEV && MACH_DAVINCI_DM6467_EVM
+   ...
```

drivers/media/video/davinci/Makefile

```
+obj-$(CONFIG_CAPTURE_DAVINCI_DM646X_EVM) +=  \
    vpif_capture.o
```

Passos, L., Guo, J., Teixeira, L., Czarnecki, K., Wąsowski, A., & Borba, P. (2013, August). Coevolution of variability models and related artifacts: a case study from the Linux kernel. In *Proceedings of the 17th International Software Product Line Conference* (pp. 91-100). ACM.

	eCos	Linux kernel	Debian	Eclipse	Android
Variability representation					
Asset base					
Basic units	Files	Files, kernel modules	Packages	Bundles	Apps
Composite units	Packages	N/A	Meta packages	Features	N/A
Unit parameters	Preproc. symbols	Preproc. symbols	Debconf options	properties/preferences	Preferences
Variability model	Feature-model-like	Feature-model-like	N/A	N/A	
Features	Packages, components, options, interfaces	Configs, choices, menuconfigs, menus	N/A	N/A	N/A
Language	CDL	Kconfig	N/A	N/A	N/A
Manifest (Schema)	N/A	N/A	y (textual DSL)	y (OSGI manifest)	y (XML-based DSL)
Grouping and categorization	Variability model	Variability model	Tasks, sections, debtags	Market place categories	App store categories
Decisions					
Decision lifecycle	Derivation	Derivation, re-config.	Re-configuration	Re-configuration	Re-configuration
Decision binding	Static	Static & dynamic	Dynamic	Dynamic	Dynamic
Derivation/re-config. tools	Configurator (ConfigTool), build system	Configurator (Kconfig), build system (Kbuild)	Installers (apt, dpkg)	Installer, market place client (P2)	Installer app (e.g., Market)
Encapsulation					
Interface mechanisms	C header files	C header files	Package-specific	Java interfaces and OSGI manifest	Explicit public components, predef. data formats
Interface specification	Documented interfaces for components, e.g., drivers	Documented interfaces for components, e.g., drivers	Package-specific, documented policies for some domains	Explicit public interfaces defined by OSGI manifest	Explicit public components, predef. data formats
Interactions					
Managed by run-time system	N/A	N/A	N/A	Equinox OSGI	Dalvik VM
Interaction mechanisms	Static linking	Static & dynamic linking	dpkg-triggers, documented policies	Class reference, services, extension points	Intent mechanism
Interaction binding	Early static	Early static & dynamic	Not specified	Late static & dynamic	Late dynamic
Platform openness	Predominantly closed	Predominantly closed	Open	Open	Open

Berger, T., Pfeiffer, R. H., Tartler, R., Dienst, S., Czarnecki, K., Wąsowski, A., & She, S. (2014). Variability mechanisms in software ecosystems. *Information and Software Technology*, *56*(11), 1520-1535.