Understanding Software Variability in Software Ecosystems

Krzysztof Czarnecki

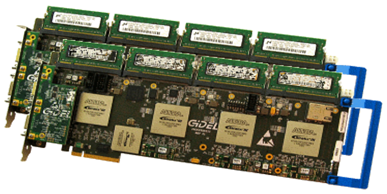
University of Waterloo

Canada

http://gsd.uwaterloo.ca

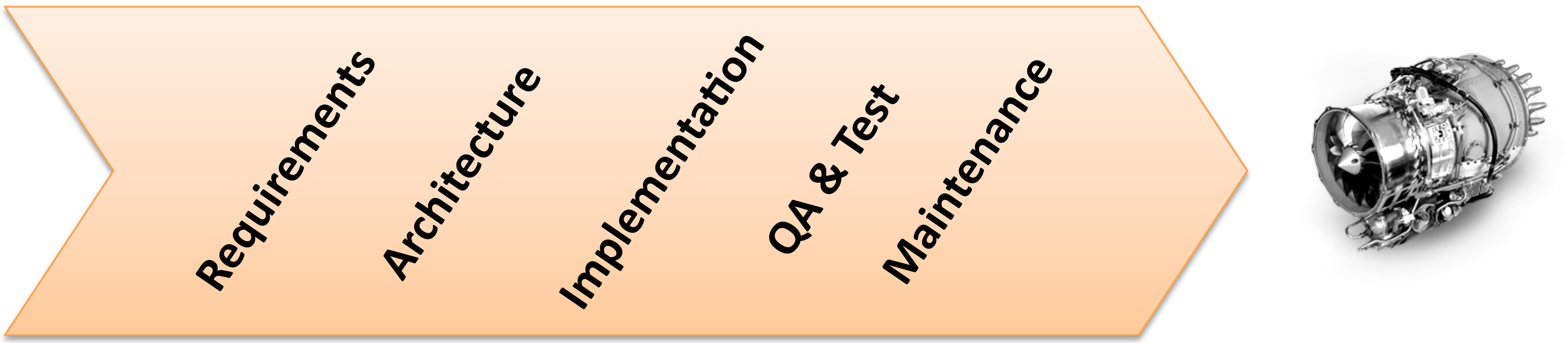
Joint work with

Thorsten Berger (U Leipzig)
Steffen Dienst (U Leipzig)
Rafael Lotufo (U Waterloo)
Marko Novakovic (U Waterloo)
Leonardo Passos (U Waterloo)
R.-Helge Pfeiffer (ITU)
Steven She (U Waterloo)
Reinhard Tartler (U Erlangen)
Andrzej Wasowski (ITU)
Yingfei Xiong (U Peking)



Variability is everywhere

Variability adds complexity



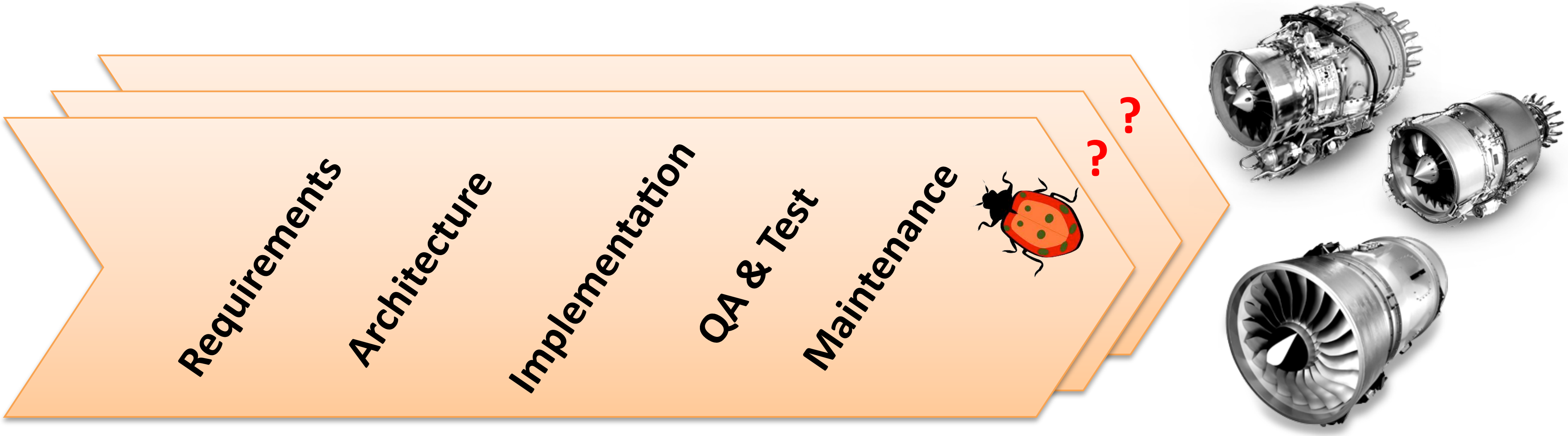
Requirements

Architecture

Implementation

QA & Test

Maintenance



Requirements

Architecture

Implementation

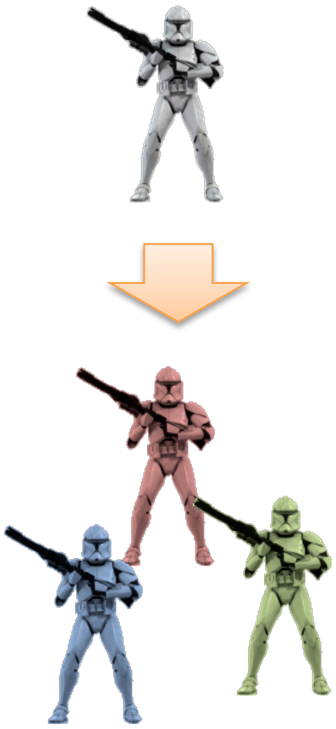
QA & Test

Maintenance

Variability Handling

Build independently

Clone & own



Share assets

Software Product Lines

- Variability modeling
- Product configuration
- Components
- Services
- DSLs
- Generators
- Design patterns
- …

Share assets

Which variability techniques are effective in practice? Under what conditions?

Variability Handling
in
Open Source Software Ecosystems

Linux Kernel

eCos

Debian

Eclipse

Android

Software ecosystems are **communities** of **developers** and **users** supported by a shared **platform** and **building on each others solutions**

Introductions…

Linux Kernel

General-purpose OS kernel

eCos

Embedded OS

Debian

Complete OS plus apps

Eclipse

Customizable IDEs

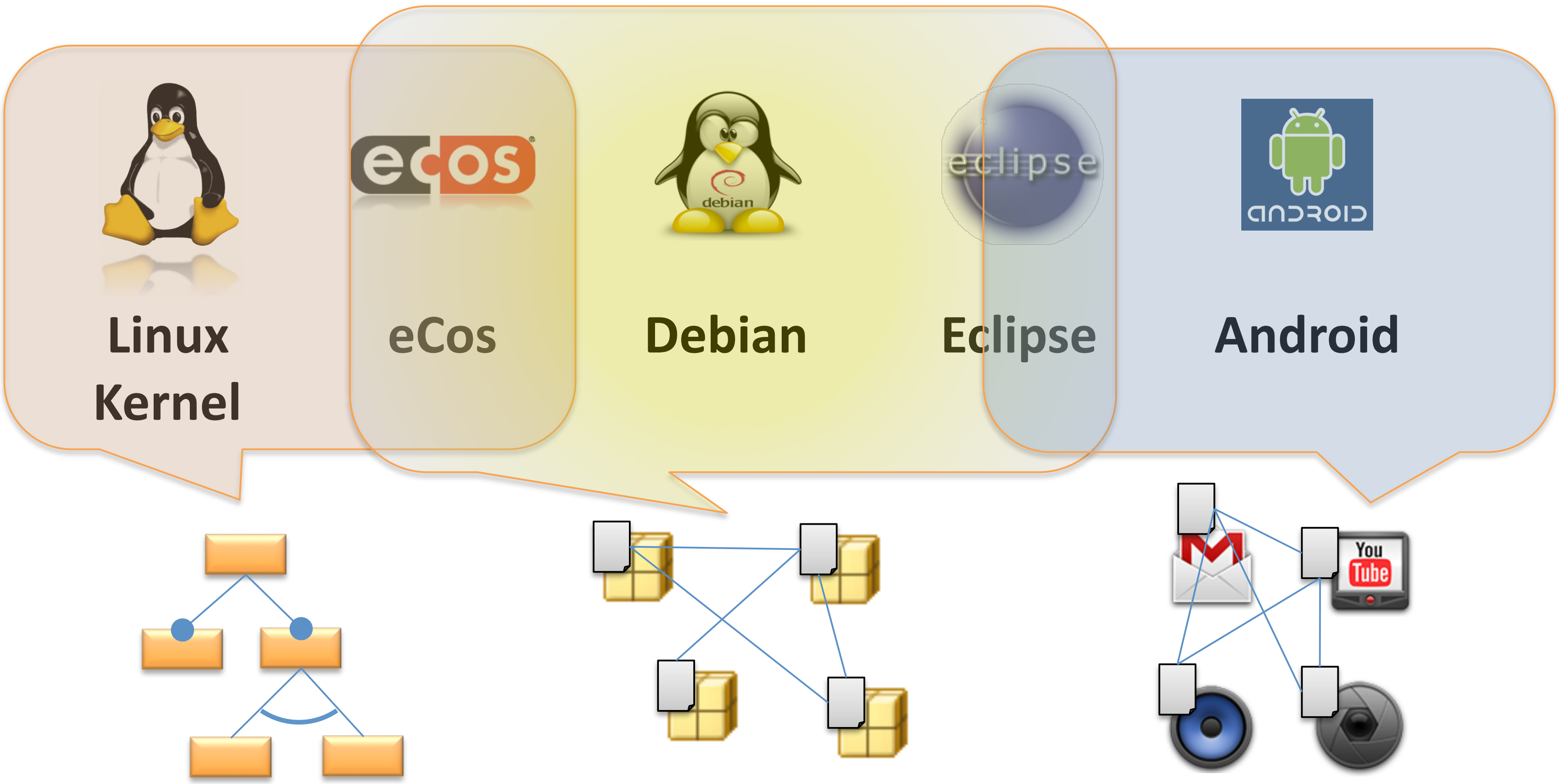
Android

Mobile OS plus apps

Each one has vast variability

Linux Kernel	eCos	Debian	Eclipse	Android
10k	2.8k	28k	>7k	>400k
Config options	Config options	Packages	Bundles	Apps

Each handles variability differently



Linux Kernel

eCos

Debian

Eclipse

Android

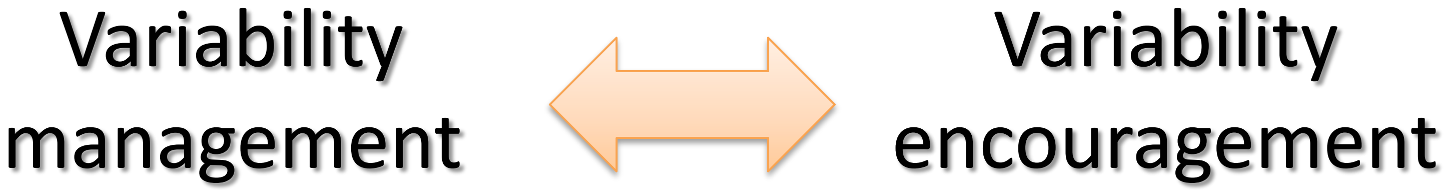
Feature models

Manifests-based packages

Service-oriented apps

Key Findings

Two Opposing Approaches to Variability



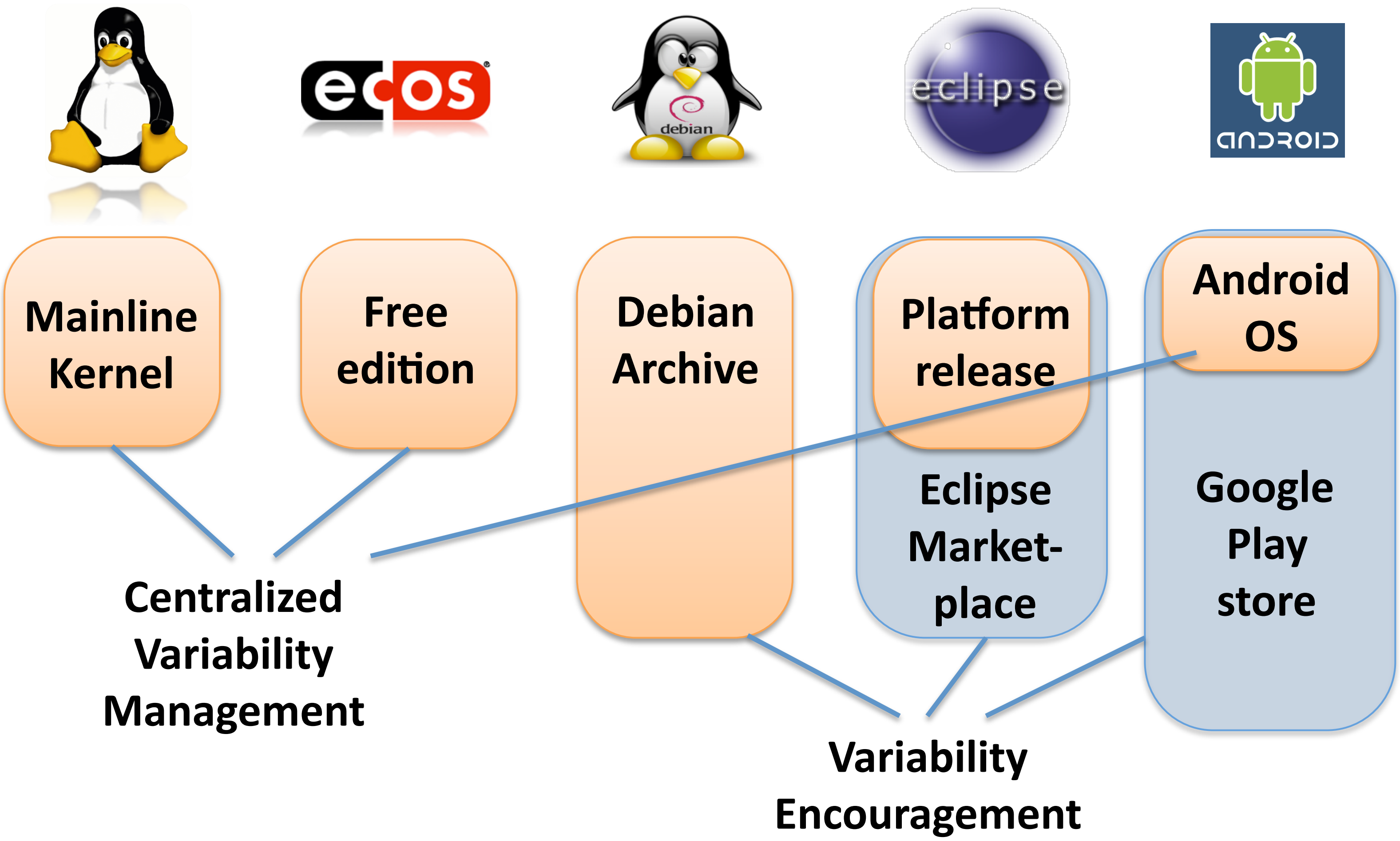
Variability management

Variability encouragement

Controlling scope
Eliminating variants if no significant business value added

Unleashing community innovation
Encouraging competition
Letting community decide scope

Variability Management vs. Encouragement



Mainline Kernel

Free edition

Debian Archive

Platform release

Eclipse Market-place

Android OS

Google Play store

Centralized Variability Management

Variability Encouragement

Feature Models

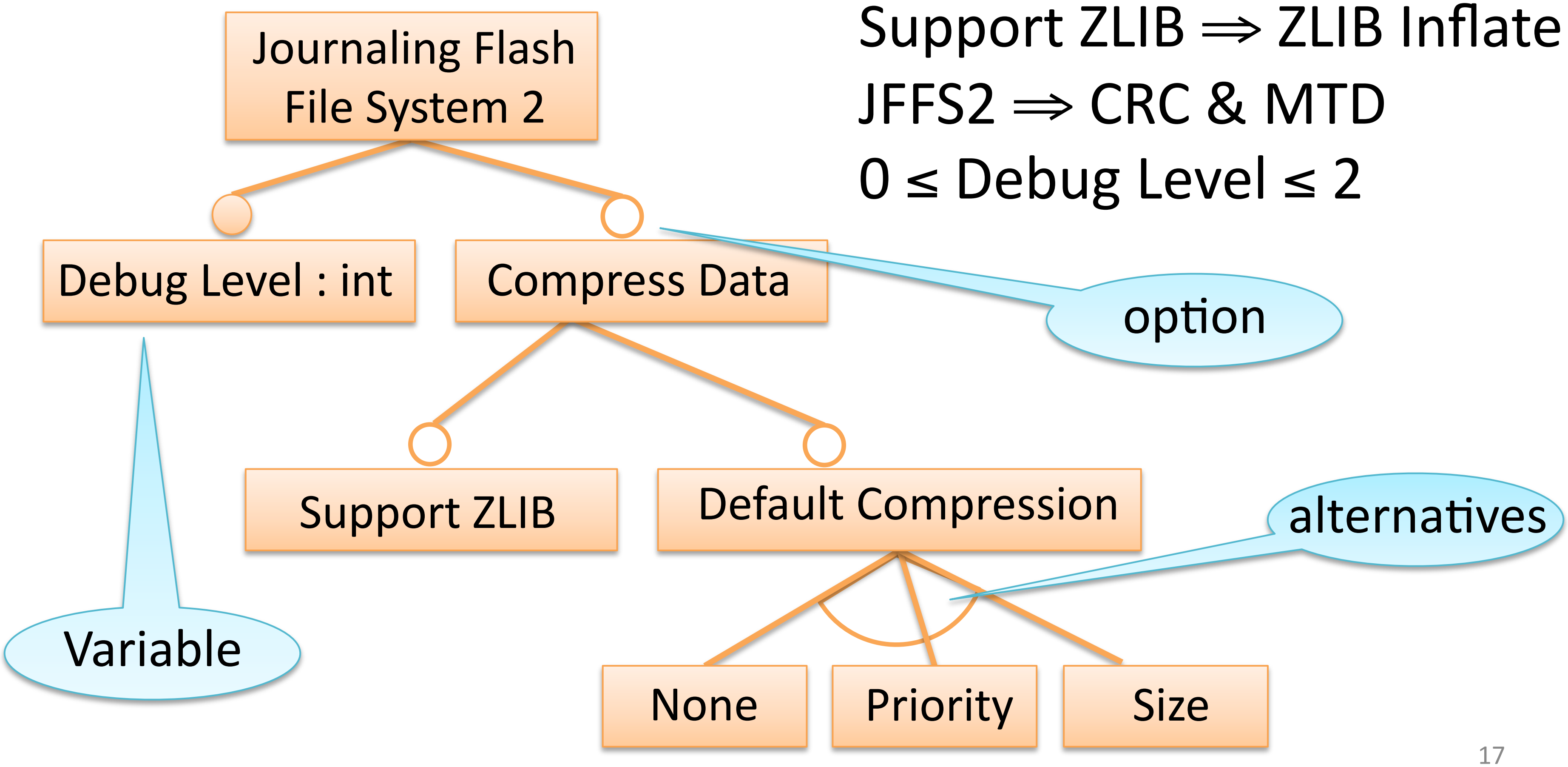
Proposed by Kang et al., 1990

A. van Deursen and P. Klint. Domain-Specific Language Design Requires Feature Descriptions, JCIT, 2002

Google Scholar has over 3k papers on feature modeling

Feature Models in Practice

Developed independently in open source – Linux, eCos

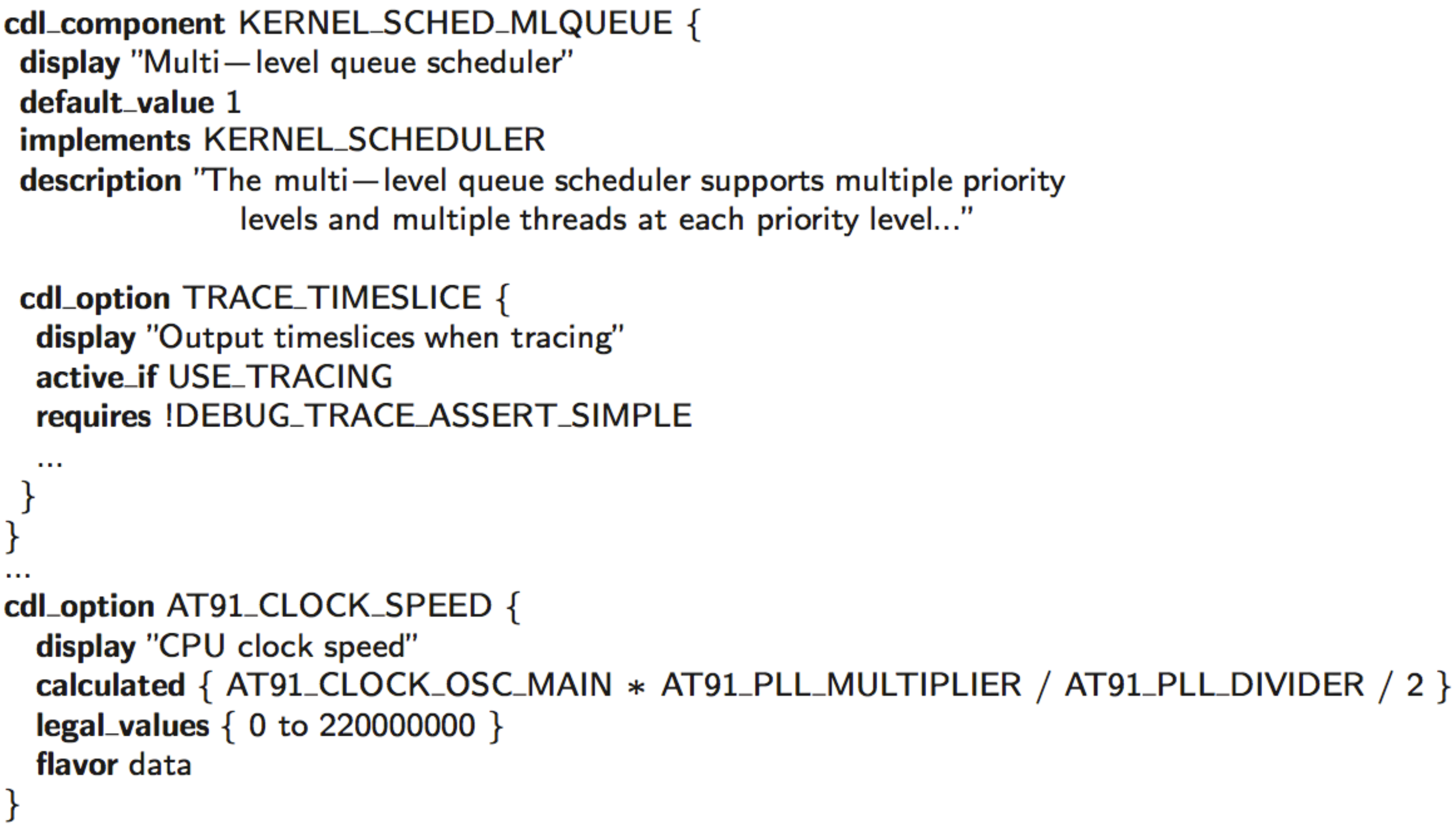


Support ZLIB ⇒ ZLIB Inflate

JFFS2 ⇒ CRC & MTD

$0 \leq \text{Debug Level} \leq 2$

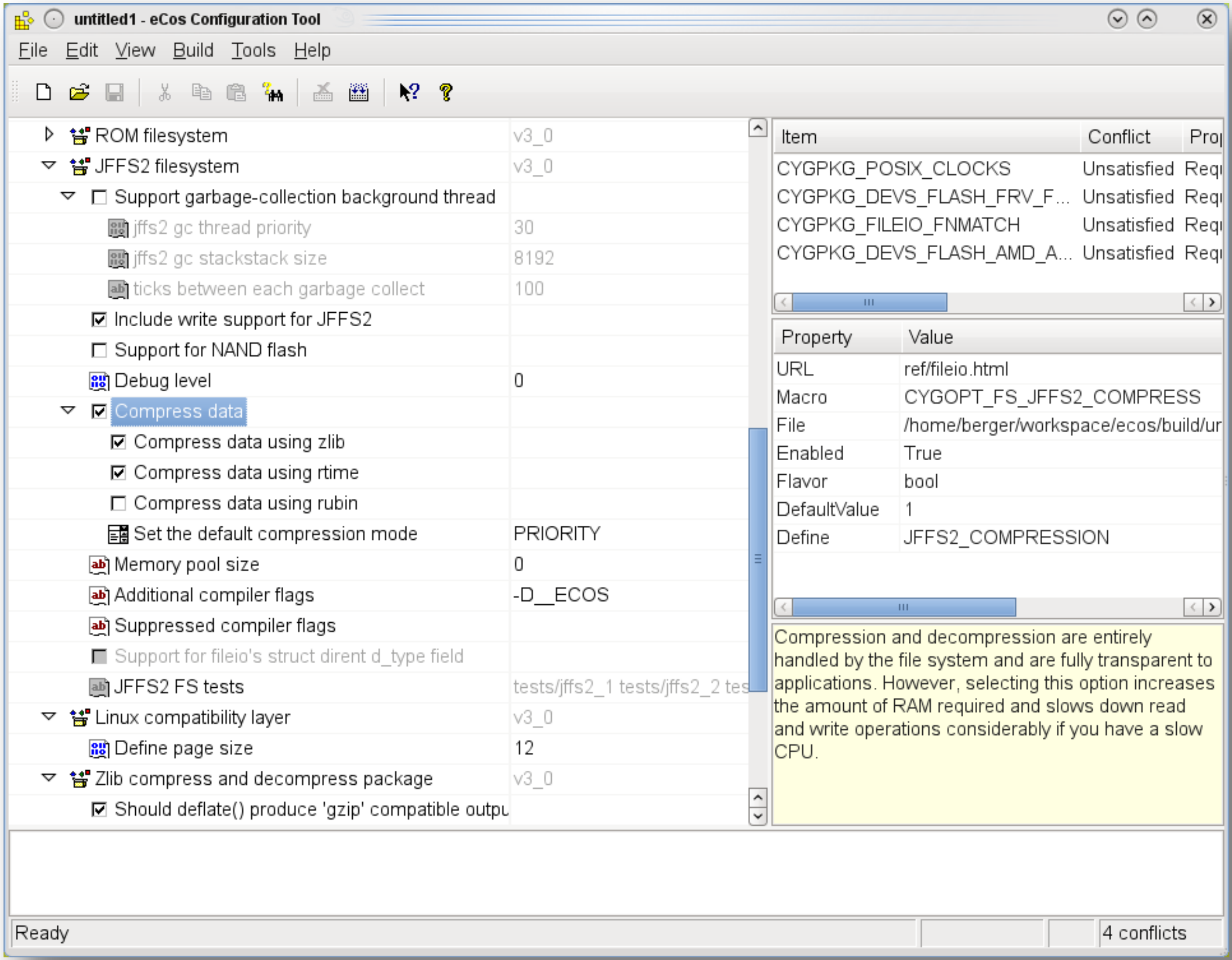
CDL Snippet



```
cdl_component KERNEL_SCHED_MLQUEUE {
 display "Multi—level queue scheduler"
 default_value 1
 implements KERNEL_SCHEDULER
 description "The multi—level queue scheduler supports multiple priority
              levels and multiple threads at each priority level..."

 cdl_option TRACE_TIMESLICE {
  display "Output timeslices when tracing"
  active_if USE_TRACING
  requires !DEBUG_TRACE_ASSERT_SIMPLE
  ...
 }
}
...
cdl_option AT91_CLOCK_SPEED {
  display "CPU clock speed"
  calculated { AT91_CLOCK_OSC_MAIN ∗ AT91_PLL_MULTIPLIER / AT91_PLL_DIVIDER / 2 }
  legal_values { 0 to 220000000 }
  flavor data
}
```

eCos Configurator



untitled1 - eCos Configuration Tool

▷ ROM filesystem	v3_0
▽ JFFS2 filesystem	v3_0
▽ ☐ Support garbage-collection background thread	
jffs2 gc thread priority	30
jffs2 gc stackstack size	8192
ticks between each garbage collect	100
☑ Include write support for JFFS2	
☐ Support for NAND flash	
Debug level	0
▽ ☑ Compress data	
☑ Compress data using zlib	
☑ Compress data using rtime	
☐ Compress data using rubin	
Set the default compression mode	PRIORITY
Memory pool size	0
Additional compiler flags	-D__ECOS
Suppressed compiler flags	
☐ Support for fileio's struct dirent d_type field	
JFFS2 FS tests	tests/jffs2_1 tests/jffs2_2 tes
▽ Linux compatibility layer	v3_0
Define page size	12
▽ Zlib compress and decompress package	v3_0
☑ Should deflate() produce 'gzip' compatible outpu	

Item	Conflict	Pro
CYGPKG_POSIX_CLOCKS	Unsatisfied	Req
CYGPKG_DEVS_FLASH_FRV_F...	Unsatisfied	Req
CYGPKG_FILEIO_FNMATCH	Unsatisfied	Req
CYGPKG_DEVS_FLASH_AMD_A...	Unsatisfied	Req

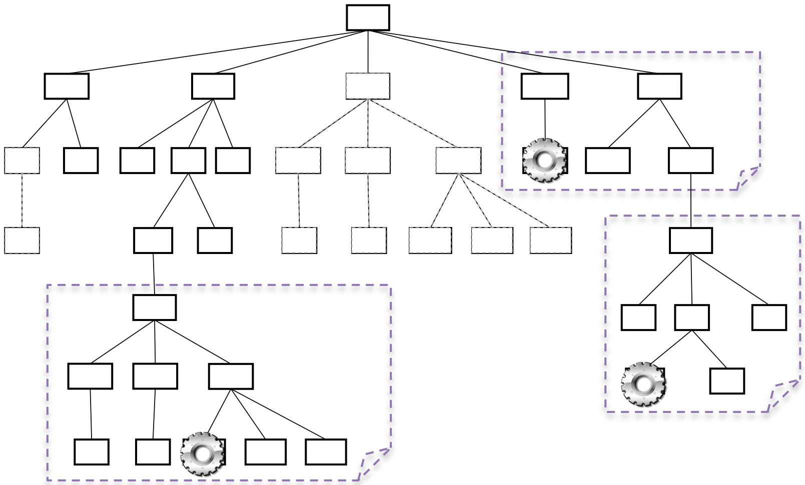
Property	Value
URL	ref/fileio.html
Macro	CYGOPT_FS_JFFS2_COMPRESS
File	/home/berger/workspace/ecos/build/ur
Enabled	True
Flavor	bool
DefaultValue	1
Define	JFFS2_COMPRESSION

Compression and decompression are entirely handled by the file system and are fully transparent to applications. However, selecting this option increases the amount of RAM required and slows down read and write operations considerably if you have a slow CPU.

Ready

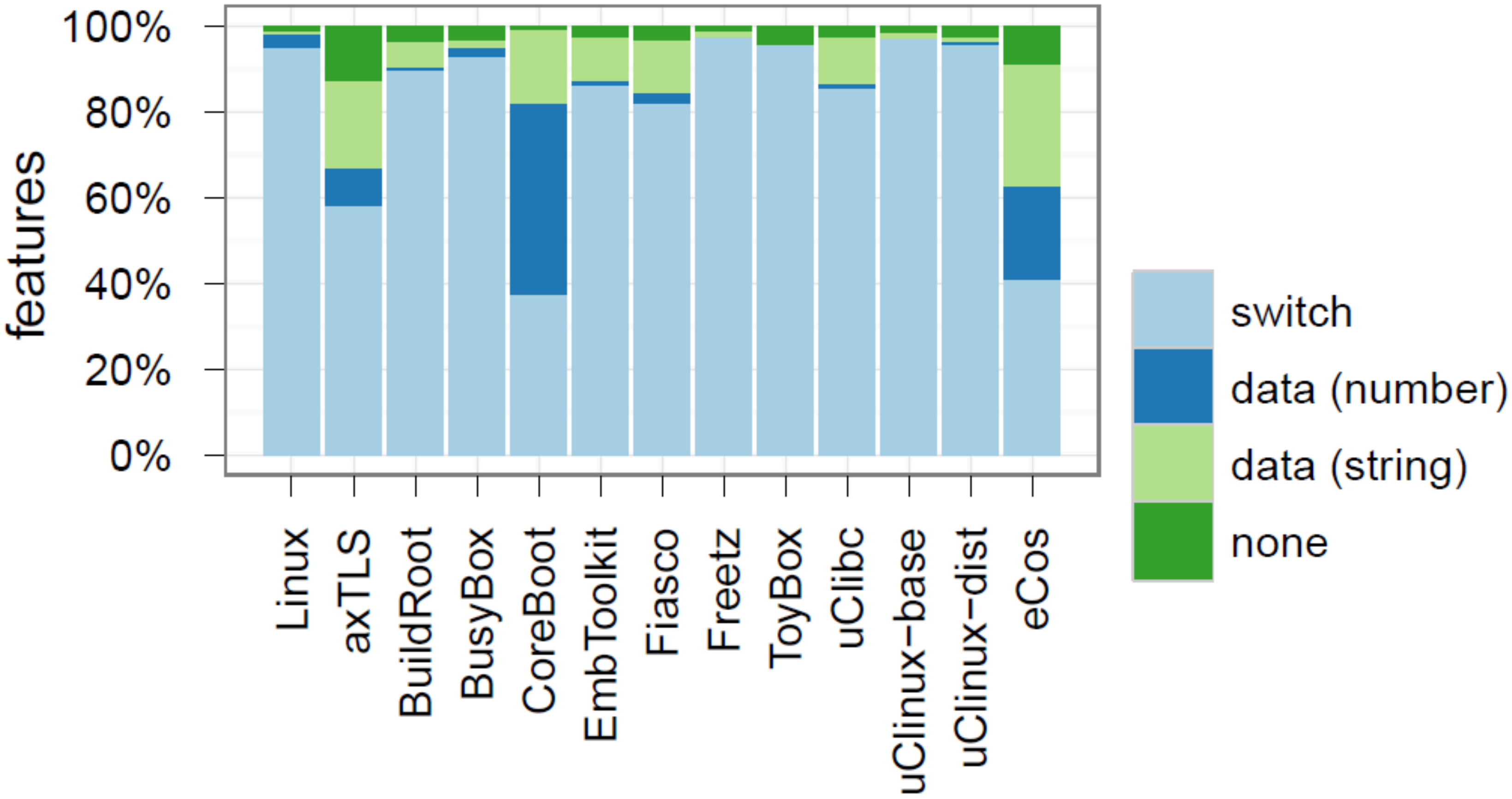
4 conflicts

Beyond Feature Modeling

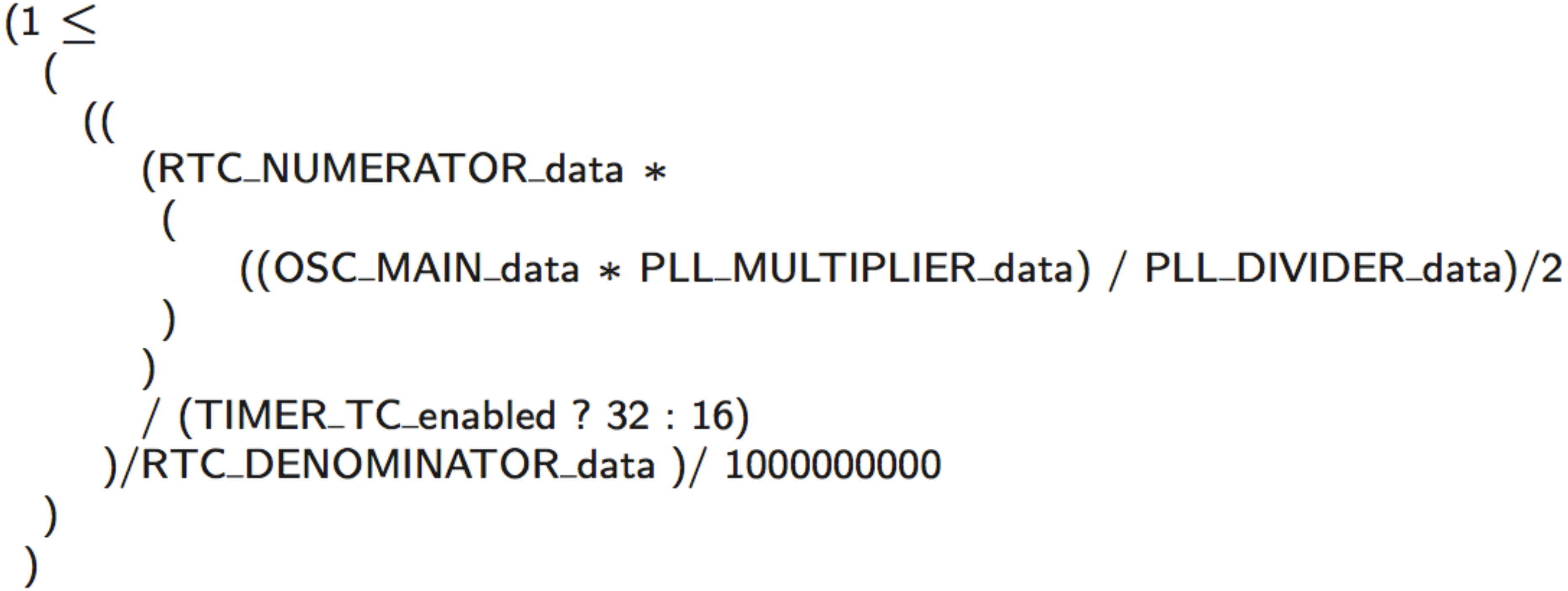


- Concepts for scalability
  - Visibility
  - Modularization
  - Derived defaults / derived features
- Expressive constraints
  - Kconfig: Three-state logic (follows Kleene‘s rules) for binding mode
  - CDL: Comparison, arithmetic and String operators
- Domain-specific vocabulary
  - E.g., package, component, option

Feature Types in Systems Software



Non-Boolean Constraints in eCos

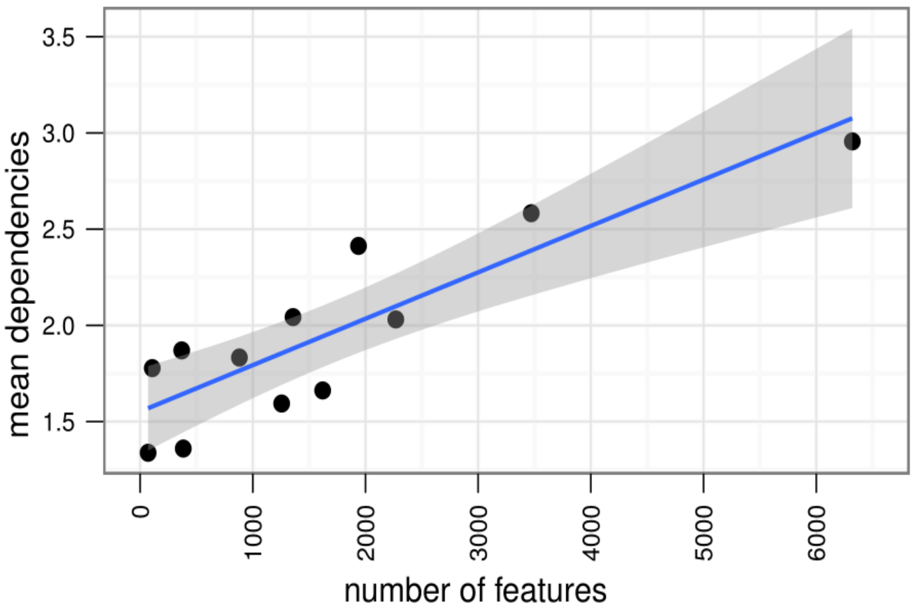


```
(1 ≤
  (
    ((
      (RTC_NUMERATOR_data ∗
       (
          ((OSC_MAIN_data ∗ PLL_MULTIPLIER_data) / PLL_DIVIDER_data)/2
       )
      )
      / (TIMER_TC_enabled ? 32 : 16)
    )/RTC_DENOMINATOR_data )/ 1000000000
  )
 )
```

=> Implications for configuration and analysis tools

Dependency Structures

Dependencies grow linearly with model size
(eCos, Linux, and 10 other Kconfig-based systems)



Indicates that the feature-based architectures scale well

Correlations

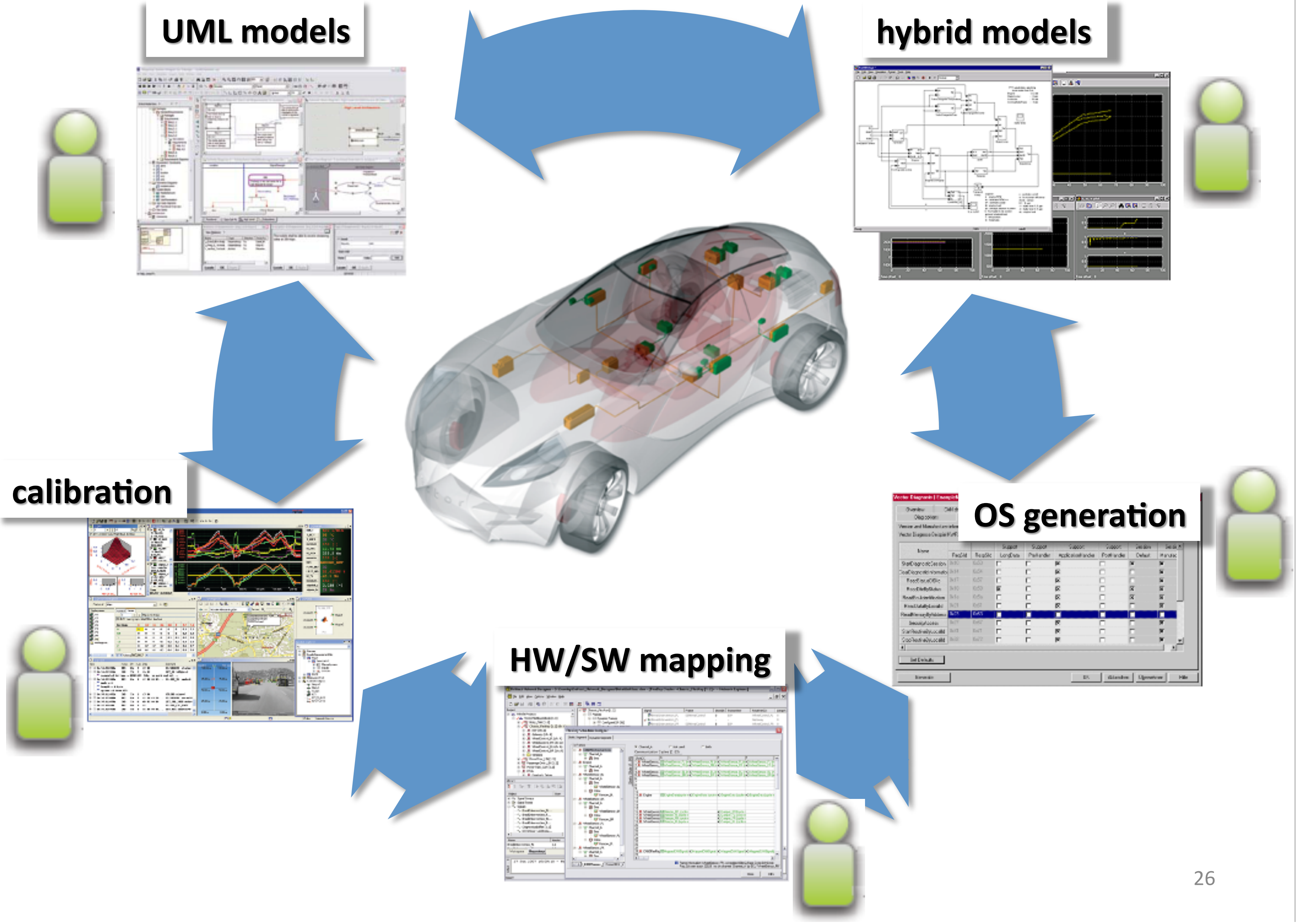
ecos
debian
eclipse
android

Highly technical ↔ End user

Static, closed config ↔ Dynamic, open config

Variability Mgmt. ↔ Variability Egmnt.

Feature modeling works well for static variability in engineering domains



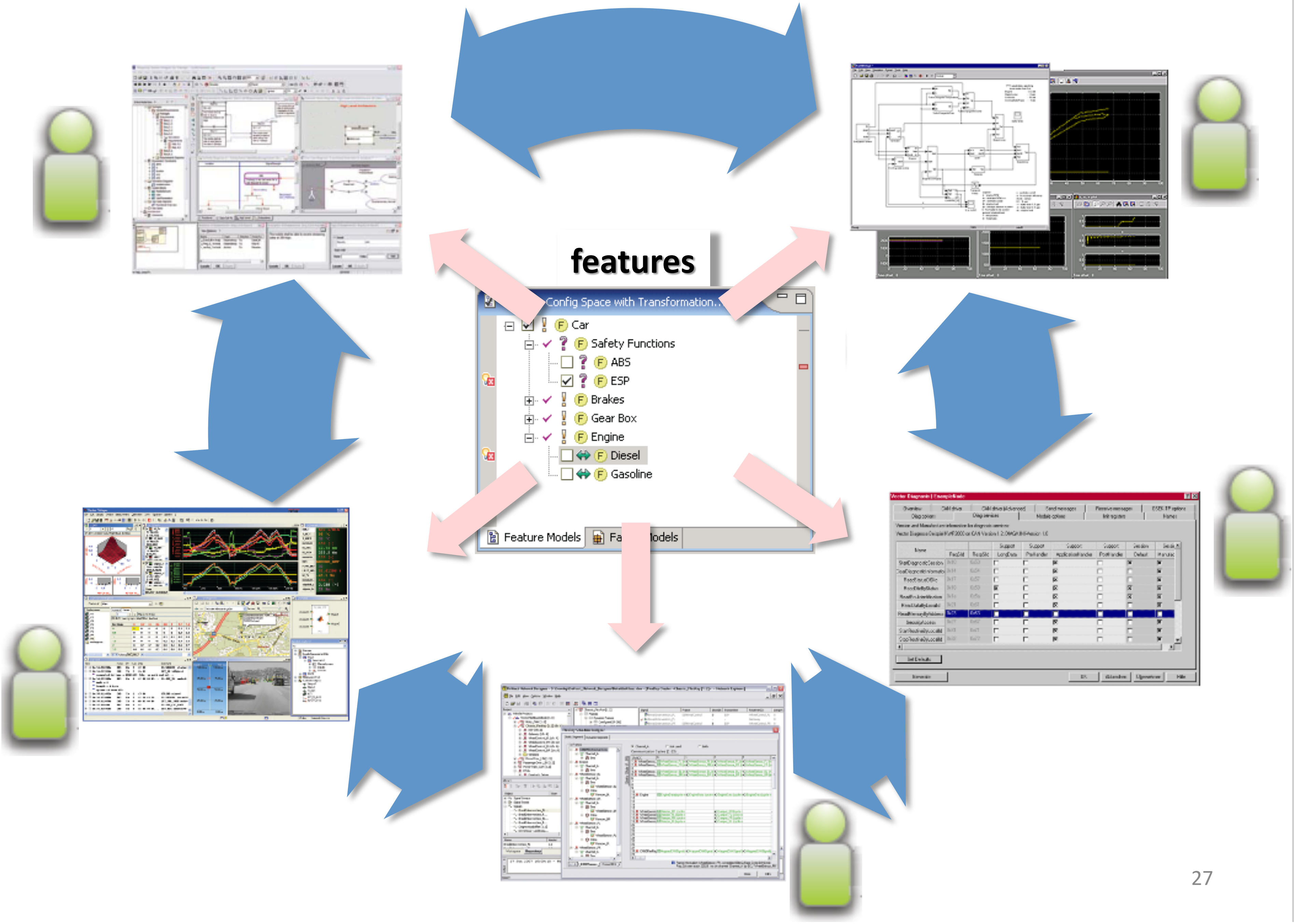
UML models

hybrid models

calibration

OS generation

HW/SW mapping



features

Config Space with Transformation...

- ☑ Car
  - ✓ Safety Functions
    - ☐ ABS
    - ☑ ESP
  - ✓ Brakes
  - ✓ Gear Box
  - ✓ Engine
    - ☐ Diesel
    - ☐ Gasoline

Variability
Management vs. Encouragement

See papers for details…

Berger, T., S. She, R. Lotufo, A. Wasowski, K. Czarnecki. Variability Modeling in the Systems Software Domain.” Submitted to TSE, 2012

Berger, T., H. Pfeiffer, R. Tartler, S.Dienst, K. Czarnecki, A.Wasowski, and Steven She. “Variability Mechanisms in Software Ecosystems.”Under submission, 2012

Passos, L., M. Novakovic, Y. Xiong, T. Berger, K. Czarnecki, and A. Wasowski, "A Study of Non-Boolean Constraints in Variability Models of an Embedded Operating System", FOSD'11, Munich, Germany, ACM, 08/2011

Hubaux, A., Y. Xiong, and K. Czarnecki, Configuration Challenges in Linux and eCos: A Survey, , no. GSDLAB-TR 2011-09-29, Waterloo, Generative Software Development Laboratory, University of Waterloo, 2011

Berger, T., S. She, R. Lotufo, A. Wasowski, and K. Czarnecki, "Variability Modeling in the Real: A Perspective from the Operating Systems Domain", 25th IEEE/ACM International Conference on Automated Software Engineering, 09/2010

She, S., R. Lotufo, T. Berger, A. Wasowski, and K. Czarnecki, "Variability Model of the Linux Kernel", Fourth International Workshop on Variability Modeling of Software-intensive Systems (VaMoS 2010), Linz, Austria, 2010

Lotufo, R., S. She, T. Berger, A. Wasowski, and K. Czarnecki, "Evolution of the Linux Kernel Variability Model", Software Product Line Conference, 09/2010

See gsd.uwaterloo.ca

Summary

Feature models seem to scale well in the embedded domain

Open and dynamic ecosystems grow fast!

They rely on dynamic binding, runtime-service lookup, and easy download and installation

Variability encouragement complements variability management as a future direction