The Variability Model of the Linux Kernel

Steven She, Rafael Lotufo, Thorsten Berger, Krzysztof Czarnecki, Andrzej Wąsowski

University of Waterloo
University of Leipzig
IT University of Copenhagen

January 26, 2010
Motivation

There is a growing number of variability modeling tools. We need **practical benchmarks** to evaluate these tools.
Motivation

There is a growing number of variability modeling tools. We need **practical benchmarks** to evaluate these tools.

Linux is a great example of a real software product line with a explicit variability model!
### Linux configurator

<table>
<thead>
<tr>
<th>Option</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAVE_ARCH_EARLY_PFN_TO_NID</td>
<td>HAVE_ARCH_EARLY_PF</td>
</tr>
<tr>
<td><strong>Power management and ACPI options</strong></td>
<td></td>
</tr>
<tr>
<td>ARCH_HIBERNATION_HEADER</td>
<td>ARCH_HIBERNATION_HEADER_HPM</td>
</tr>
<tr>
<td><strong>Power Management support</strong></td>
<td>PM</td>
</tr>
<tr>
<td>PM_TRACE</td>
<td>PM_TRACE</td>
</tr>
<tr>
<td>Suspend/resume event tracing</td>
<td>PM_TRACE_RTC</td>
</tr>
<tr>
<td>PM_SLEEP_SMP</td>
<td>PM_SLEEP_SMP</td>
</tr>
<tr>
<td>PM_SLEEP</td>
<td>PM_SLEEP</td>
</tr>
<tr>
<td><strong>Suspend to RAM and standby</strong></td>
<td>SUSPEND</td>
</tr>
<tr>
<td><strong>Hibernation (aka 'suspend to disk')</strong></td>
<td>HIBERNATION</td>
</tr>
<tr>
<td>Advanced Power Management Emulation</td>
<td>APM_EMULATION</td>
</tr>
<tr>
<td><strong>ACPI (Advanced Configuration and Power Interface) Support</strong></td>
<td>ACPI</td>
</tr>
<tr>
<td>X86_APM_BOOT</td>
<td>X86_APM_BOOT</td>
</tr>
<tr>
<td><strong>APM (Advanced Power Management) BIOS support</strong></td>
<td>APM</td>
</tr>
<tr>
<td>CPU Frequency scaling</td>
<td></td>
</tr>
</tbody>
</table>

Variability model is specified using the **Kconfig** language.
menu "Power management and ACPI options"
   depends on !X86_VOYAGER

config PM
   bool "Power Management support"
   depends on !IA64_HP_SIM
   ---help---
   "Power Management" means that ...

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

config CPU_IDLE
   bool "CPU idle PM support"
   default ACPI
endmenu
Kconfig declaration

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

config PM
  bool "Power Management support"
  depends on !IA64_HP_SIM
  ---help---
    "Power Management" means that ...

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

config CPU_IDLE
  bool "CPU idle PM support"
  default ACPI
endmenu
Kconfig declaration

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

config PM
   bool "Power Management support"
   depends on !!A64_HP_SIM
   ---help---
   "Power Management" means that ...

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

config CPU_IDLE
   bool "CPU idle PM support"
   default ACPI
endmenu
Kconfig declaration

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

config PM
  bool "Power Management support"
  depends on !IA64_HP_SIM
  ---help---
    "Power Management" means that ...

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

config CPU_IDLE
  bool "CPU idle PM support"
  default ACPI

endmenu
```
Kconfig declaration

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

config PM
  bool "Power Management support"
depends on !IA64_HP_SIM
  ---help---
  "Power Management" means that ...

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

config CPU_IDLE
  bool "CPU idle PM support"
default ACPI
endmenu
Kconfig declaration

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

config PM
  bool "Power Management support"
  depends on !IA64_HP_SIM
  ---help---
      "Power Management" means that ...

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

config CPU_IDLE
  bool "CPU idle PM support"
  default ACPI
endmenu
```
Kconfig feature model

Linux 2.6.28.6 Configuration

Power Management and ACPI Options

PM

PM_DEBUG

CPU_IDLE

PM_MENU → X86_Voyager
PM → IA64_HP_SIM
ACPI ↔ CPU_IDLE
Linux KConfig $\rightarrow$ Feature Model

Analyzed four aspects of the Linux 2.6.28.6 Kconfig model in terms of feature modeling concepts:

- characterized features,
- model hierarchy,
- constraints,
- and natural language properties.
Comparing with published models

Compared Linux statistics with 32 published models\(^1\).

- 19 models - software product lines
- 8 models - other product lines (e.g. hardware, business)
- 5 models - domain models (e.g. eCommerce systems)

Only 5 models describe real, existing software systems.

\(^1\)http://www.splot-research.org
Size of published models

Linux Kconfig model has 5426 features.
Size of published models

Linux Kconfig model has 5426 features.
### Linux feature statistics

<table>
<thead>
<tr>
<th>Kconfig Concept</th>
<th>Features</th>
<th>Mand.</th>
<th>Grouped</th>
<th>XOR + OR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Config</strong></td>
<td>5323</td>
<td>0</td>
<td>146</td>
<td>0</td>
</tr>
<tr>
<td>Non / User-Sel.</td>
<td>547 + 4744</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td>2005</td>
<td>0</td>
<td>136</td>
<td>0</td>
</tr>
<tr>
<td>Tristate</td>
<td>3130</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Int</td>
<td>132</td>
<td>132</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hex</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>String</td>
<td>27</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Menu</strong></td>
<td>71</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Choice</strong></td>
<td>32</td>
<td>31</td>
<td>0</td>
<td>30 + 2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5426</td>
<td>257</td>
<td>146</td>
<td>30 + 2</td>
</tr>
</tbody>
</table>
# Linux feature statistics

<table>
<thead>
<tr>
<th>Kconfig Concept</th>
<th>Features</th>
<th>Mand.</th>
<th>Grouped</th>
<th>XOR + OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config</td>
<td>5323</td>
<td>0</td>
<td>146</td>
<td>0</td>
</tr>
<tr>
<td>Non / User-Sel.</td>
<td>547 + 4744</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td>2005</td>
<td>0</td>
<td>136</td>
<td>0</td>
</tr>
<tr>
<td>Tristate</td>
<td>3130</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Int</td>
<td>132</td>
<td>132</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hex</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>String</td>
<td>27</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Menu</td>
<td>71</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Choice</td>
<td>32</td>
<td>31</td>
<td>0</td>
<td>30 + 2</td>
</tr>
<tr>
<td>Total</td>
<td>5426</td>
<td>257</td>
<td>146</td>
<td>30 + 2</td>
</tr>
</tbody>
</table>
## Linux feature statistics

<table>
<thead>
<tr>
<th>Kconfig Concept</th>
<th>Features</th>
<th>Mand.</th>
<th>Grouped</th>
<th>XOR + OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config</td>
<td>5323</td>
<td>0</td>
<td>146</td>
<td>0</td>
</tr>
<tr>
<td>Non / User-Sel.</td>
<td>547 + 4744</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td>2005</td>
<td>0</td>
<td>136</td>
<td>0</td>
</tr>
<tr>
<td>Tristate</td>
<td>3130</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Int</td>
<td>132</td>
<td>132</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hex</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>String</td>
<td>27</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Menu</td>
<td>71</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Choice</td>
<td>32</td>
<td>31</td>
<td>0</td>
<td>30 + 2</td>
</tr>
<tr>
<td>Total</td>
<td>5426</td>
<td>257</td>
<td>146</td>
<td>30 + 2</td>
</tr>
</tbody>
</table>
# Linux feature statistics

<table>
<thead>
<tr>
<th>Kconfig Concept</th>
<th>Features</th>
<th>Mand.</th>
<th>Grouped</th>
<th>XOR + OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config</td>
<td>5323</td>
<td>0</td>
<td>146</td>
<td>0</td>
</tr>
<tr>
<td>Non / User-Sel.</td>
<td>547 + 4744</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Boolean</strong></td>
<td>2005</td>
<td>0</td>
<td>136</td>
<td>0</td>
</tr>
<tr>
<td><strong>Tristate</strong></td>
<td>3130</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td><strong>Int</strong></td>
<td>132</td>
<td>132</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Hex</strong></td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td>27</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Menu</strong></td>
<td>71</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Choice</strong></td>
<td>32</td>
<td>31</td>
<td>0</td>
<td>30 + 2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5426</td>
<td>257</td>
<td>146</td>
<td>30 + 2</td>
</tr>
</tbody>
</table>
## Linux feature statistics

<table>
<thead>
<tr>
<th>Kconfig Concept</th>
<th>Features</th>
<th>Mand.</th>
<th>Grouped</th>
<th>XOR + OR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Config</strong></td>
<td>5323</td>
<td>0</td>
<td>146</td>
<td>0</td>
</tr>
<tr>
<td><strong>Non / User-Sel.</strong></td>
<td>547 + 4744</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td>2005</td>
<td>0</td>
<td>136</td>
<td>0</td>
</tr>
<tr>
<td>Tristate</td>
<td>3130</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Int</td>
<td>132</td>
<td>132</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hex</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>String</td>
<td>27</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Menu</strong></td>
<td>71</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Choice</strong></td>
<td>32</td>
<td>31</td>
<td>0</td>
<td>30 + 2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5426</td>
<td>257</td>
<td>146</td>
<td>30 + 2</td>
</tr>
</tbody>
</table>
# Linux feature statistics

<table>
<thead>
<tr>
<th>Kconfig Concept</th>
<th>Features</th>
<th>Mand.</th>
<th>Grouped</th>
<th>XOR + OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config</td>
<td>5323</td>
<td>0</td>
<td>146</td>
<td>0</td>
</tr>
<tr>
<td>Non / User-Sel.</td>
<td>547 + 4744</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td>2005</td>
<td>0</td>
<td>136</td>
<td>0</td>
</tr>
<tr>
<td>Tristate</td>
<td>3130</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Int</td>
<td>132</td>
<td>132</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hex</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>String</td>
<td>27</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Menu</strong></td>
<td>71</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Choice</strong></td>
<td>32</td>
<td>31</td>
<td>0</td>
<td>30 + 2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5426</td>
<td>257</td>
<td>146</td>
<td>30 + 2</td>
</tr>
</tbody>
</table>
# Linux feature statistics

<table>
<thead>
<tr>
<th>Kconfig Concept</th>
<th>Features</th>
<th>Mand.</th>
<th>Grouped</th>
<th>XOR + OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config</td>
<td>5323</td>
<td>0</td>
<td>146</td>
<td>0</td>
</tr>
<tr>
<td>Non / User-Sel.</td>
<td>547 + 4744</td>
<td>0</td>
<td>146</td>
<td>0</td>
</tr>
<tr>
<td>Boolean</td>
<td>2005</td>
<td>0</td>
<td>136</td>
<td>0</td>
</tr>
<tr>
<td>Tristate</td>
<td>3130</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Int</td>
<td>132</td>
<td>132</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hex</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>String</td>
<td>27</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Menu</td>
<td>71</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Choice</td>
<td>32</td>
<td>31</td>
<td>0</td>
<td>30 + 2</td>
</tr>
<tr>
<td>Total</td>
<td>5426</td>
<td>257</td>
<td>146</td>
<td>30 + 2</td>
</tr>
</tbody>
</table>
### Linux feature statistics

<table>
<thead>
<tr>
<th>Kconfig Concept</th>
<th>Features</th>
<th>Mand.</th>
<th>Grouped</th>
<th>XOR + OR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Config</strong></td>
<td>5323</td>
<td>0</td>
<td>146</td>
<td>0</td>
</tr>
<tr>
<td><strong>Non / User-Sel.</strong></td>
<td>547 + 4744</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boolean</td>
<td>2005</td>
<td>0</td>
<td>136</td>
<td>0</td>
</tr>
<tr>
<td>Tristate</td>
<td>3130</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Int</td>
<td>132</td>
<td>132</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hex</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>String</td>
<td>27</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Menu</strong></td>
<td>71</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Choice</strong></td>
<td>32</td>
<td>31</td>
<td>0</td>
<td>30 + 2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>5426</td>
<td>257</td>
<td>146</td>
<td>30 + 2</td>
</tr>
</tbody>
</table>
## Published models vs. Linux

<table>
<thead>
<tr>
<th>Concept</th>
<th>Published Models (%)</th>
<th>Linux (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>median</td>
<td>min</td>
</tr>
<tr>
<td>mandatory features</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>grouped features</td>
<td>44</td>
<td>0</td>
</tr>
<tr>
<td>groups</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>XOR</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>OR</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>
Code-granularity of features

Two heuristics for automatic feature selection in the Linux configurator: *allyes*, *allno*.

<table>
<thead>
<tr>
<th>Metric</th>
<th>allyes</th>
<th>allno</th>
<th>$\Delta$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>3,448</td>
<td>61</td>
<td>3387</td>
<td>1</td>
</tr>
<tr>
<td>Files</td>
<td>10,326</td>
<td>973</td>
<td>9,353</td>
<td>2.76</td>
</tr>
<tr>
<td>SLOC</td>
<td>4,266,171</td>
<td>210,302</td>
<td>4,055,869</td>
<td>1,197.48</td>
</tr>
</tbody>
</table>

$\Delta_i = \text{allyes}_i - \text{allno}_i; \theta_i = \Delta_i/\Delta_1$
# Code-granularity of features

Two heuristics for automatic feature selection in the Linux configurator: *allyes, allno*.

<table>
<thead>
<tr>
<th>Metric</th>
<th>allyes</th>
<th>allno</th>
<th>Δ</th>
<th>θ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>3,448</td>
<td>61</td>
<td>3387</td>
<td>1</td>
</tr>
<tr>
<td>Files</td>
<td>10,326</td>
<td>973</td>
<td>9,353</td>
<td>2.76</td>
</tr>
<tr>
<td>SLOC</td>
<td>4,266,171</td>
<td>210,302</td>
<td>4,055,869</td>
<td>1,197.48</td>
</tr>
</tbody>
</table>

$$\Delta_i = \text{allyes}_i - \text{allno}_i; \ \theta_i = \Delta_i / \Delta_1$$
Qualitative characteristics

Subjective categorization of 180 randomly selected features.

<table>
<thead>
<tr>
<th>Menu</th>
<th>Support</th>
<th>Option</th>
<th>Debug</th>
<th>No Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>97</td>
<td>46</td>
<td>13</td>
<td>23</td>
</tr>
</tbody>
</table>

User-based granularity categories of Linux features

<table>
<thead>
<tr>
<th>API</th>
<th>Driver</th>
<th>Kernel</th>
<th>Protocol</th>
<th>Subsystem</th>
<th>No Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>120</td>
<td>15</td>
<td>14</td>
<td>1</td>
<td>25</td>
</tr>
</tbody>
</table>

Types of features in Linux kernel
### Qualitative characteristics

Subjective categorization of 180 randomly selected features.

<table>
<thead>
<tr>
<th></th>
<th>Menu</th>
<th>Support</th>
<th>Option</th>
<th>Debug</th>
<th>No Info</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>97</td>
<td>46</td>
<td>13</td>
<td>23</td>
</tr>
</tbody>
</table>

User-based granularity categories of Linux features

<table>
<thead>
<tr>
<th></th>
<th>API</th>
<th>Driver</th>
<th>Kernel</th>
<th>Protocol</th>
<th>Subsystem</th>
<th>No Info</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>120</td>
<td>15</td>
<td>14</td>
<td>1</td>
<td>25</td>
</tr>
</tbody>
</table>

Types of features in Linux kernel
Hierarchy statistics

- 4544 leaves
- well-balanced
- relatively shallow
- many single childs
- long tail
Constraint statistics

- 9291 constraints
- 82% features referenced
- 89% requires constraints
- some v. large constraints

Significant number of cross-tree constraints!
Constraint statistics

- 9291 constraints
- 82% features referenced
- 89% requires constraints
- some v. large constraints

Significant number of cross-tree constraints!
Identifiers, prompts and descriptions

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

"Power Management" means that parts of your computer are shut off or put into a power conserving "sleep" mode if they are not being used. There are two competing standards for doing this: APM and ACPI. If you want to use either one, say Y here and then also to the requisite support below.

Power Management is most important for battery powered laptop computers; if you have a laptop, check out the Linux Laptop home page on the WWW at <http://www.linux-on-laptops.com/> or Tuxmobil - Linux on Mobile Computers at <http://www.tuxmobil.org/> and the Battery Powered Linux mini-HOWTO, available from <http://www.tldp.org/docs.html#howto>.
Natural language properties

<table>
<thead>
<tr>
<th>artifact</th>
<th>no. of characters</th>
<th>no. of words</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>median</td>
<td>min</td>
</tr>
<tr>
<td>identifiers</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>prompts</td>
<td>27</td>
<td>2</td>
</tr>
<tr>
<td>description</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Size of textual artifacts

<table>
<thead>
<tr>
<th>text source</th>
<th>most frequent domain terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>usb snd md serial fb debug</td>
</tr>
<tr>
<td>prompt</td>
<td>usb ethernet pci intel scsi pcmcia</td>
</tr>
<tr>
<td>description</td>
<td>usb linux scsi ethernet pci howto</td>
</tr>
</tbody>
</table>
# Natural language properties

<table>
<thead>
<tr>
<th>artifact</th>
<th>no. of characters</th>
<th>no. of words</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>median</td>
<td>min</td>
</tr>
<tr>
<td>identifiers</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>prompts</td>
<td>27</td>
<td>2</td>
</tr>
<tr>
<td>description</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

## Size of textual artifacts

<table>
<thead>
<tr>
<th>text source</th>
<th>most frequent domain terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>usb, snd, md, serial, fb, debug</td>
</tr>
<tr>
<td>prompt</td>
<td>usb, ethernet, pci, intel, scsi, pcmcia</td>
</tr>
<tr>
<td>description</td>
<td>usb, linux, scsi, ethernet, pci, howto</td>
</tr>
</tbody>
</table>

---

The Variability Model of the Linux Kernel
Related work

- Sincero and Schröder-Preikschat (ASPL 2008)
- Tartler et al. - dead feature detection in Linux (FOSD 2009)
- Segura and Ruiz-Cortés - FM benchmarks (VaMoS 2009)
Conclusions

- Low number of mandatory features and groups.
- Each feature crosses roughly 2.8 source files and 1200 SLOC.
- Average leaf depth of 4, many single childs and long tail.
- Significant number and size of cross-tree constraints.

\(^1\)Available for download at http://fm.gsdlab.org
Counting Constraints

Counted the number of distinct literals involved in the declaration of a constraint.

\((\textit{MWINCHIP3D} \lor \textit{gcMCRUSOE} \lor \textit{gMEFFICEON} \lor \textit{MCYRIXIII}) \lor \textit{MK7} \lor \textit{MK6} \lor \textit{MPENTIUM4} \lor \textit{MPENTIUMM} \lor \textit{MPENTIUMIII}) \lor \textit{MPENTIUMIII} \lor \textit{M686} \lor \textit{M586MMX} \lor \textit{M586TSC} \lor \textit{MK8}) \lor \textit{MVIAC3_2} \lor \textit{MVIAC7} \lor \textit{MGEODEGX1} \lor \textit{MGEODE_LX} \lor \textit{MCORE2}) \land \neg \textit{X86_NUMAQ} \lor \textit{X86_64} \rightarrow \textit{X86_TSC} = y\)

Figure: Equation with 22 distinct literals
The Variability Model of the Linux Kernel
## Kconfig → Feature Model mapping

<table>
<thead>
<tr>
<th>concepts</th>
<th>Feature modeling concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch config</td>
<td>Optional feature</td>
</tr>
<tr>
<td>Entry-field config</td>
<td>Mandatory feature</td>
</tr>
<tr>
<td>Conditional menu</td>
<td>Optional feature</td>
</tr>
<tr>
<td>Unconditional menu</td>
<td>Mandatory feature</td>
</tr>
<tr>
<td>Choice</td>
<td></td>
</tr>
<tr>
<td>Mandatory</td>
<td>Mandatory feat. + xor-group</td>
</tr>
<tr>
<td>Optional</td>
<td>Optional feat. + xor-group</td>
</tr>
<tr>
<td>Mandatory tristate</td>
<td>Mandatory feat. + or-group</td>
</tr>
<tr>
<td>Optional tristate</td>
<td>Optional feat. + or-group</td>
</tr>
<tr>
<td>Choice config</td>
<td>Grouped feature</td>
</tr>
<tr>
<td>Config, menu or choice nesting</td>
<td></td>
</tr>
<tr>
<td>Visibility conditions</td>
<td></td>
</tr>
<tr>
<td>Selects</td>
<td>Cross-tree constraint</td>
</tr>
<tr>
<td>Constraining defaults</td>
<td></td>
</tr>
</tbody>
</table>