Variability Modeling in the Real:
A Perspective from the Operating Systems Domain

25th IEEE/ACM International Conference on Automated Software Engineering
ASE 2010
Antwerp, Belgium, Sept. 22nd, 2010

Thorsten Berger, Steven She, Rafael Lotufo,
Andrzej Wasowski, Krzysztof Czarnecki
Biologists go into the Wild for their Studies…
Biologists go into the Wild for their Studies…
Biologists go into the Wild for their Studies…
Biologists go into the Wild for their Studies…

Feature Model

- **Lion**
  - **Sex**
    - Male
    - Female
  - **Habitat**
  - **Mane**
  - **Appearance**
  - **Color**
  - **Size**
    - Small
    - Medium
    - Large
Variability Modeling

Thorsten Berger
Variability Modeling

Thorsten Berger
Variability Modeling

Configuration #1

Variability Model
(15810 possible configurations)
Variability Modeling

Configuration #1

Variability Model
(15810 possible configurations)

Configuration #2
“Variability needs in software are constantly increasing, because:

- Variability moves from mechanics and hardware to software,
- Design decisions are delayed as long as economically feasible.”

Jan Bosch
20 Years of Feature Models

Jubilee

Celebratee  Event  Hangover

Nice  Remarkable

Congratulations to
Kyo C. Kang
VaMoS 2010
Linz, Austria
Other Domains

20 Years of Feature Models

Jubilee
Celebratee  Event  Hangover

Nice  Remarkable

Congratulations to
Kyo C. Kang
VaMoS 2010
Linz, Austria
Feature Modeling

Other Domains

Tools

20 Years of Feature Models

Jubilee
Celebratee Event Hangover
Nice Remarkable

Congratulations to
Kyo C. Kang
VaMoS 2010
Linz, Austria

Thorsten Berger
Feature Modeling

Other Domains

Standardization Efforts

Tools

Common Variability Language (CVL)

eclipse.org/proposals/feature-model

Thorsten Berger
NOW, WHAT‘S THE PROBLEM?
What about Empirical Research?

- A lot of research on *variability*, but not on *real models*!
What about Empirical Research?

- A lot of research on *variability*, but not on *real models*!
- Some *industry reports* available, but *not* the models!
What about Empirical Research?

- A lot of research on *variability*, but not on **real models**!
- Some *industry reports* available, but not **the models**!
- **Assumptions** of real models used for *synthetic* examples!
What about Empirical Research?

- A lot of research on variability, but not on real models!
- Some industry reports available, but not the models!
- Assumptions of real models used for synthetic examples!
- Literature studies:
What about Empirical Research?

- A lot of research on variability, but not on real models!
- Some industry reports available, but not the models!
- Assumptions of real models used for synthetic examples!
- Literature studies:
  - Hubaux et al. [VAMOS10]:
    - Only 2% of reviewed papers (8 of 415) discuss applications of Feature Models in practice
    - Few details about their usage given
What about Empirical Research?

- A lot of research on variability, but not on real models!
- Some industry reports available, but not the models!
- Assumptions of real models used for synthetic examples!
- Literature studies:
  - *Hubaux et al. [VAMOS10]*:
    - Only 2% of reviewed papers (8 of 415) discuss applications of Feature Models in practice
    - Few details about their usage given
  - *Chen et al. [SPLC09]*:
    - “There is only little, if any, experimental or detailed comparative analysis … of different VM approaches.”
    - All VM approaches share similar concepts
    - Some sort of reference model needed for model transformations, tools and future research
OUR STUDY...
Three Research Questions

1. Can we provide quantitative and qualitative empirical evidence *whether* Feature Modeling concepts are used in real-world languages?

2. Are additional concepts needed?

3. Are the assumptions about real models in the literature correct?
Three Research Questions

1. Can we provide quantitative and qualitative empirical evidence whether Feature Modeling concepts are used in real-world languages?

2. Are additional concepts needed?

3. Are the assumptions about real models in the literature correct?
Three Research Questions

1. Can we provide quantitative and qualitative empirical evidence \textit{whether} Feature Modeling concepts are used in real-world languages?

2. Are additional concepts needed?

3. Are the assumptions about real models in the literature correct?
We study

CONCEPTS, SEMANTICS AND USAGE OF…
Kconfig and CDL

**Kconfig language**

- **Linux Kernel 2.6.32**
  - (22 hardware architectures, 6.4 mio. SLOC)
  - 6320 Features (X86)

**CDL language**

- **eCos 3.0**
  - (embedded RTOS, 116 architectures, ~1 mio. SLOC)
  - 1244 Features (I386)
Kconfig and CDL

Kconfig language

Linux Kernel 2.6.32
(22 hardware architectures, 6.4 mio. SLOC)

6320 Features (X86)
Kconfig and CDL

CDL language

eCos 3.0
(embedded RTOS,
116 architectures,
~1 mio. SLOC)

244 Features (I386)
Kconfig and CDL

Kconfig language

```
menuconfig MISC_FILESYSTEMS
  bool "Miscellaneous filesystems"
  if MISC_FILESYSTEMS
    config JFFS2_FS
    tristate "Joumalling Flash File System" if MTD
    select CRC32 if MTD
    config JFFS2_FS_DEBUG
    int "JFFS2 Debug level (0=quiet, 2=noisy)"
    depends on JFFS2_FS
    default 6
    range 0, 2
    --- help ---
    Debug verbosity of ...
    config JFFS2_FS_WRITEBUFFER
    bool
    depends on JFFS2_FS
    default HAS IOMEM
    config JFFS2_COMPRESS
    bool "Advanced compression options for JFFS2"
    depends on JFFS2_FS
    config JFFS2_ZLIB
    bool "Compress w/ zlib..." if JFFS2_COMPRESS
    depends on JFFS2_FS
    select ZLIB_INFLATE
    default y
    choice
    prompt "Default compression" if JFFS2_COMPRESS
    default JFFS2_COMPRESS_NONE
    depends on JFFS2_FS
    config JFFS2_COMPRESS_NONE
    bool "no compression"
    config JFFS2_COMPRESS_NONE
    bool "Priority"
    config JFFS2_COMPRESS_SIZE
    bool "size (EXPERIMENTAL)"
    endchoice
  endif
```
What do we mean by variability model

SEMANTICS?
Variability Model Semantics

- Configuration Space Semantics
Variability Model Semantics

- Configuration Space Semantics

Kconfig Model

```
menuconfig MISC_FILESYSTEMS
  bool "Miscellaneous filesystems"
  if MISC_FILESYSTEMS
    config JFFS2_FS
      tristate "Journalling Flash FS" if MTD
      select CRC32 if MTD
    config JFFS2_FS_DEBUG
      int "JFFS2 Debug level (0=quiet, 2=noisy)"
      depends on JFFS2_FS
      default 0
      range 0 2
      --- help ---
      Debug verbosity of ...
```

\[ kconfig : Kconfig \to \mathcal{P}(\text{Conf}) \]

\[ C_1 = \{(\text{JFFS2}, y), (\text{JFFS2_DEBUG}, 2), \ldots\} \]

\[ C_2 = \{(\text{JFFS2}, m), (\text{JFFS2_DEBUG}, 0), \ldots\} \]

\[ C_n = \{(\text{JFFS2}, n), (\text{JFFS2_DEBUG}, 0), \ldots\} \]
METHODOLOGY
Methodology - Design Space

- Reverse-engineered formal semantics (denotational style)
- Reverse-engineered formal semantics (denotational style)
- Reverse-engineered formal semantics (denotational style)
- Reverse-engineered formal semantics (denotational style)

- Abstract Syntax
- Semantic Function
- Semantic Domain

Propositional Semantic Function
Propositional Semantic Domain

Propositional Abstraction for Reasoners

Examples
Tools and their source code
Documentation
extension of configurators

LibCDL

XConfig
Methodology - Language Usage

- Extension of configurators
- Linux and eCos models

LibCDL: model export

LibCDL: Intermediate Model

XConfig: model export

Kconfig: Intermediate Model
Methodology - Language Usage

- **extension of configurators**
- **Linux and eCos models**
- **analysis infrastructure**

**LibCDL**
- model export
- model
- CDL
  - Intermediate Model
  - parsing

**XConfig**
- model export
- model
- Kconfig
  - Intermediate Model
  - parsing
Methodology - Language Usage

- Extension of configurators
- Linux and eCos models
- Analysis infrastructure

LibCDL

XConfig

CDL: Intermediate Model

Kconfig: Intermediate Model

Model export

Parsing

Statistics

Thorsten Berger
Methodology - Language Usage

extension of configurators

Linux and eCos models

analysis infrastructure

propositional abstraction

propositional abstraction

LibCDL

CDL

Intermediate Model

model export

parsing

xConfig

Kconfig

Intermediate Model

model export

parsing

ϕ

ecos

number of features

number of models

ϕ

linux

Number of referenced features

Transformation

Statistics

Transformation

Statistics
Methodology - Language Usage

extension of configurators

Linux and eCos models

analysis infrastructure

propositional abstraction

Transformation

Statistics

LibCDL

model export

CDL
Intermediate Model

parsing

XConfig

model export

Kconfig
Intermediate Model

parsing

analysis infrastructure

Transformation

Statistics

SAT-based analysis

propositional abstraction

Linux and eCos models

propositional abstraction

ϕ
ecos

ϕ
linux
RESULTS
Three Research Questions

1. Can we provide quantitative and qualitative empirical evidence whether Feature Modeling concepts are used in real-world languages?

2. Are additional concepts needed?

3. Are the assumptions about real models in the literature correct?
Three Research Questions

1. Can we provide quantitative and qualitative empirical evidence whether Feature Modeling concepts are used in real-world languages?

2. Are additional concepts needed?

3. Are the assumptions about real models in the literature correct?
Feature Modeling Concepts

- Concepts:
  - Boolean (optional), Integer and String features (attributes)
  - Hierarchy
  - Group constraints
  - Cross-tree constraints
Feature Modeling Concepts

- Concepts:
  - Boolean (optional), Integer and String features (attributes)
  - Hierarchy
  - Group constraints
  - Cross-tree constraints

- But intricate semantic interactions
  - Needed many iterations to get the formal semantics right
  - Often slightly different from what we expected
Feature Modeling Concepts

- Concepts:
  - Boolean (optional), Integer and String features (attributes)
  - Hierarchy
  - Group constraints
  - Cross-tree constraints

- But intricate semantic interactions
  - Needed many iterations to get the formal semantics right
  - Often slightly different from what we expected

- Some (but minor) violations of feature modeling rules
Feature Modeling Concepts

- Concepts:
  - Boolean (optional), Integer and String features (attributes)
  - Hierarchy
  - Group constraints
  - Cross-tree constraints

- But intricate semantic interactions
  - Needed many iterations to get the formal semantics right
  - Often slightly different from what we expected

- Some (but minor) violations of feature modeling rules

- Languages benefit from being domain-specific
Three Research Questions

1. Can we provide quantitative and qualitative empirical evidence whether Feature Modeling concepts are used in real-world languages?

2. Are additional concepts needed?

3. Are the assumptions about real models in the literature correct?
Three Research Questions

1. Can we provide quantitative and qualitative empirical evidence whether Feature Modeling concepts are used in real-world languages?

2. Are additional concepts needed?

3. Are the assumptions about real models in the literature correct?
- Concepts for scalability
Beyond Feature Modeling

- Concepts for scalability
  - Visibility
Beyond Feature Modeling

- Concepts for scalability
  - Visibility
  - Modularization
Beyond Feature Modeling

- Concepts for scalability
  - Visibility
  - Modularization
  - Derived defaults / derived features
Beyond Feature Modeling

- Concepts for scalability
  - Visibility
  - Modularization
  - Derived defaults / derived features

- Expressive constraints
  - Kconfig: Three-state logic (follows Kleene‘s rules)
  - CDL: Comparison, arithmetic and String operators
Beyond Feature Modeling

- **Concepts for scalability**
  - Visibility
  - Modularization
  - Derived defaults / derived features

- **Expressive constraints**
  - Kconfig: Three-state logic (follows Kleene’s rules)
  - CDL: Comparison, arithmetic and String operators

- **Code mappings / build specifications**
Three Research Questions

1. Can we provide quantitative and qualitative empirical evidence whether Feature Modeling concepts are used in real-world languages?  
   YES

2. Are additional concepts needed?  
   YES

3. Are the assumptions about real models in the literature correct?
Three Research Questions

1. Can we provide quantitative and qualitative empirical evidence whether Feature Modeling concepts are used in real-world languages?

YES

2. Are additional concepts needed?

YES

3. Are the assumptions about real models in the literature correct?

NO
Assumptions

- We always see nicely balanced trees...


www.feasible.de/description/bsp_ess_en.html

code.google.com/p/dsl/variantmanagement/wiki/DemoShowCase
But Linux and eCos models are very shallow!

<table>
<thead>
<tr>
<th></th>
<th>Avg. depth</th>
<th>Max. depth</th>
<th>Max. branching</th>
<th>No. of leaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>4</td>
<td>8</td>
<td>158</td>
<td>5316 (84%)</td>
</tr>
<tr>
<td>eCos</td>
<td>3</td>
<td>6</td>
<td>29</td>
<td>947 (76%)</td>
</tr>
</tbody>
</table>
Assumption

- But Linux and eCos models are very shallow!

<table>
<thead>
<tr>
<th></th>
<th>Avg. depth</th>
<th>Max. depth</th>
<th>Max. branching</th>
<th>No. of leaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>4</td>
<td>8</td>
<td>158</td>
<td>5316 (84%)</td>
</tr>
<tr>
<td>eCos</td>
<td>3</td>
<td>6</td>
<td>29</td>
<td>947 (76%)</td>
</tr>
</tbody>
</table>
- But Linux and eCos models are very shallow!
Assumption

- But Linux and eCos models are very shallow!

<table>
<thead>
<tr>
<th></th>
<th>Avg. depth</th>
<th>Max. depth</th>
<th>Max. branching</th>
<th>No. of leaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>4</td>
<td>8</td>
<td>158</td>
<td>5316 (84%)</td>
</tr>
<tr>
<td>eCos</td>
<td>3</td>
<td>6</td>
<td>29</td>
<td>947 (76%)</td>
</tr>
</tbody>
</table>
Assumptions

- Branching factors with high variation!

<table>
<thead>
<tr>
<th></th>
<th>Avg. depth</th>
<th>Max. depth</th>
<th>Max. branching</th>
<th>No. of leaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>4</td>
<td>8</td>
<td>158</td>
<td>5316 (84%)</td>
</tr>
<tr>
<td>eCos</td>
<td>3</td>
<td>6</td>
<td>29</td>
<td>947 (76%)</td>
</tr>
</tbody>
</table>
Assumptions

- Branching factors with high variation!

<table>
<thead>
<tr>
<th></th>
<th>Avg. depth</th>
<th>Max. depth</th>
<th>Max. branching</th>
<th>No. of leaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>4</td>
<td>8</td>
<td>158</td>
<td>5316 (84%)</td>
</tr>
<tr>
<td>eCos</td>
<td>3</td>
<td>6</td>
<td>29</td>
<td>947 (76%)</td>
</tr>
</tbody>
</table>
Assumptions

- Branching factors with high variation!

<table>
<thead>
<tr>
<th></th>
<th>Avg. depth</th>
<th>Max. depth</th>
<th>Max. branching</th>
<th>No. of leaves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>4</td>
<td>8</td>
<td>158</td>
<td>5316 (84%)</td>
</tr>
<tr>
<td>eCos</td>
<td>3</td>
<td>6</td>
<td>29</td>
<td>947 (76%)</td>
</tr>
</tbody>
</table>
Some More Findings

- Cross-Tree Constraints ratio
  - Literature: 30-40%
  - Our models: 86%
Some More Findings

- Cross-Tree Constraints ratio
  - Literature: 30-40%
  - Our models: 86%
- Very large constraints exist (Linux: 56 features, eCos: 21)
Some More Findings

- Cross-Tree Constraints ratio
  - Literature: 30-40%
  - Our models: 86%
- Very large constraints exist (Linux: 56 features, eCos: 21)
- Very few group constraints (OR, XOR, MUTEX)
Some More Findings

- Cross-Tree Constraints ratio
  - Literature: 30-40%
  - Our models: 86%
- Very large constraints exist (Linux: 56 features, eCos: 21)
- Very few group constraints (OR, XOR, MUTEX)
- Kconfig and CDL configurators
  - Configuration process is *Re-Configuration*!
  - Limited or no reasoning support
    - Kconfig relies on an imperative construct for choice propagation
    - CDL has an inference engine, which is correct but incomplete
Some More Findings

- Cross-Tree Constraints ratio
  - Literature: 30-40%
  - Our models: 86%
- Very large constraints exist (Linux: 56 features, eCos: 21)
- Very few group constraints (OR, XOR, MUTEX)
- Kconfig and CDL configurators
  - Configuration process is Re-Configuration!
  - Limited or no reasoning support
    - Kconfig relies on an imperative construct for choice propagation
    - CDL has an inference engine, which is correct but incomplete
- Many more details in the paper!
CONCLUSIONS
Conclusions

- Empirical studies are fundamentally necessary in the VM field to guide future research and to provide requirements for tool developers.
Conclusions

- Empirical studies are fundamentally necessary in the VM field to guide future research and to provide requirements for tool developers.

- The more ways we look at how real languages are designed and how models look like, the more confidence we have that we understand Software Product Lines.
Conclusions

- Empirical studies are fundamentally necessary in the VM field to guide future research and to provide requirements for tool developers.

- The more ways we look at how real languages are designed and how models look like, the more confidence we have that we understand Software Product Lines.

- In studying the models in-depth, our findings have confirmed – and refuted – previous knowledge about variability languages and models.
Conclusions

- Empirical studies are fundamentally necessary in the VM field to guide future research and to provide requirements for tool developers.

- The more ways we look at how real languages are designed and how models look like, the more confidence we have that we understand Software Product Lines.

- In studying the models in-depth, our findings have confirmed – and refuted – previous knowledge about variability languages and models.

- Understanding languages and extracting these models that were evolved over 10 years kept us PhD students and the professors busy for almost half a year!
Variability Modeling in the Real: A Perspective from the Operating Systems Domain

25th IEEE/ACM International Conference on Automated Software Engineering
ASE 2010
Antwerp, Belgium, Sept. 22nd, 2010

Thorsten Berger, Steven She, Rafael Lotufo, Andrzej Wasowski, Krzysztof Czarnecki
APPENDIX
EXAMPLES
• Children can exclude their parent
- **Children can exclude their parent**

```c
config VIDEO_HELPER_CHIPS_AUTO
    bool "Autoselect pertinent encoders/decoders and other helper chips"
    default y
    ---help---
    Most video cards may require additional modules to encode or

config VIDEO_IR_I2C
    tristate "I2C module for IR" if !VIDEO_HELPER_CHIPS_AUTO
    depends on I2C && VIDEO_IR
    default y
    ---help---
    Most boards have an IR chip directly connected via GPIO.

menu "Encoders/decoders and other helper chips"
    depends on !VIDEO_HELPER_CHIPS_AUTO

comment "Audio decoders"

config VIDEO_TVAUDIO
    tristate "Simple audio decoder chips"
    depends on VIDEO_V4L2 && I2C
    ---help---
    Support for several audio decoder chips found on some bt8xx boards:
```
Defaults can impose constraints in Kconfig

```c
config DW_DMAC
    tristate "Synopsys DesignWare AHB DMA support"
    depends on AVR32
    select DMA_ENGINE
    default y if CPU_AT32AP7000
    ---help---
    Support the Synopsys DesignWare AHB DMA controller. This
can be integrated in chips such as the Atmel AT32ap7000.
```

- We thought just
  - DW_DMAC → DMA_Engine ∧ AVR32
- But instead also
  - !AVR32 ∧ CPU_AT32AP7000 → DW_DMAC ∧ DMA_ENGINE
Computed features

- Computation of test cases

```c
cdl_component CYGPKG_HAL_TESTS {
    display "Common HAL tests"
    flavor data
    no_define
    calculated { "tests/context tests/basic,
                 . ((!CYGINT_HAL_TESTS_NO_CACHES) ? " tests/cache" : "")
                 . ((CYGPKG_HAL_BUILD_COMPILER_TESTS) ? " tests/cpp1 tests/vaargs" : "")
                 . ((!CYGVAR_KERNEL_COUNTERS_CLOCK) ? " tests/intr" : "") }
    description "This option specifies the set of tests for the common HAL."
```
VARIABILITY MODELING APPROACHES
Variability Modeling Techniques

Feature Models (FODA)

- CDL
- Forfamel
- OVM
- ConIPF
- COVAMOF
- Koalish
- Kumbang
- VSL
- RequiLine
- Kumbang
- VPM
- KobrA
- FDL
- CBFM
- CONSUL
- DRM
- Gears
- Kconfig
Feature Modeling Example

- JFFS2 filesystem

Support ZLIB $\rightarrow$ ZLIB Inflate
JFFS2 $\rightarrow$ CRC $\land$ MTD
$0 \leq$ Debug Level $\leq 2$
Feature Modeling Example

- JFFS2 filesystem

```
Support ZLIB → ZLIB Inflate
JFFS2 → CRC ∧ MTD
0 ≤ Debug Level ≤ 2
```
Feature Modeling Example

- JFFS2 filesystem

**Hierarchy**

- Misc. Filesystems
  - Journalling Flash File System
    - Debug Level : Int
    - Compress Data
      - Support ZLIB
        - Default Compression
          - None
          - Priority
          - Size

Support ZLIB $\rightarrow$ ZLIB Inflate
JFFS2 $\rightarrow$ CRC $\wedge$ MTD
$0 \leq$ Debug Level $\leq$ 2
Feature Modeling Example

- JFFS2 filesystem

**Hierarchy**

- Misc. Filesystems
  - Journalling Flash File System
    - Debug Level : Int
    - Compress Data
      - Support ZLIB
      - Default Compression
        - None
        - Priority
        - Size

**Features**

- (Boolean (optional), String, Int)

**Group Constraints**

- Support ZLIB → ZLIB Inflate
- JFFS2 → CRC ∧ MTD
- 0 ≤ Debug Level ≤ 2
Feature Modeling Example

- JFFS2 filesystem

Hierarchy

Features

(Bool (optional), String, Int)

Cross-Tree Constraints

Group Constraints

Support ZLIB → ZLIB Inflate
JFFS2 → CRC ∧ MTD
0 ≤ Debug Level ≤ 2

Thorsten Berger