

Software Product Lines, Variability, and Configurations Overview and Principles

Mathieu Acher
Maître de Conférences
mathieu.acher@irisa.fr

Material

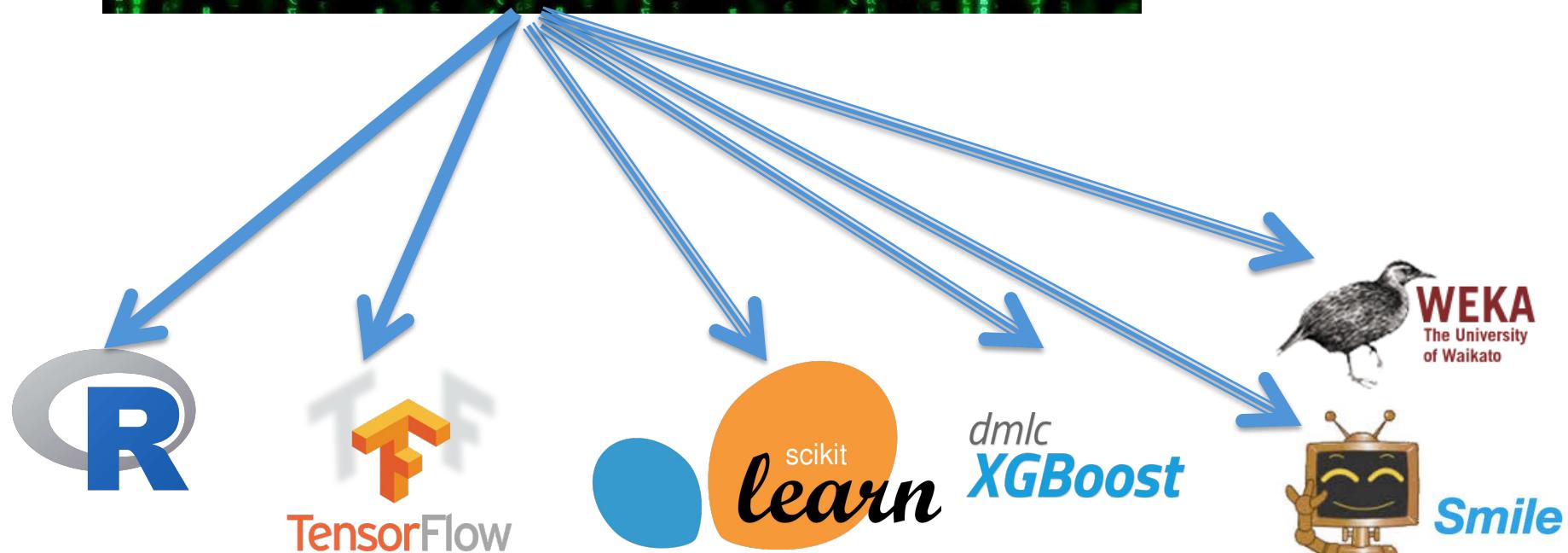
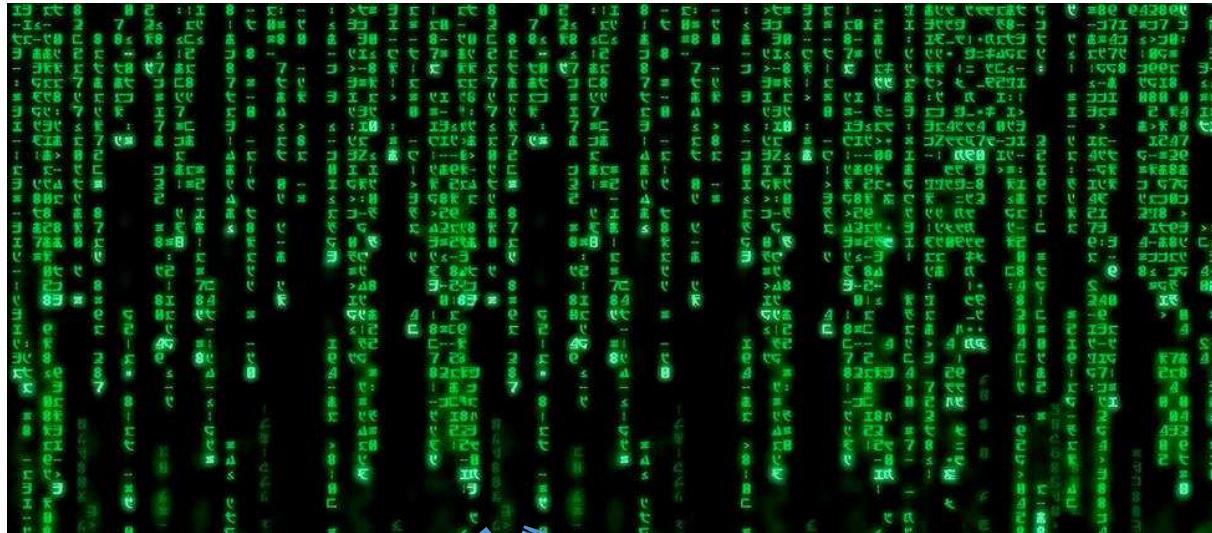
<https://github.com/acherm/teaching-MDE1920->

MIAGE

<http://teaching.variability.io>

<http://familiar.variability.io>

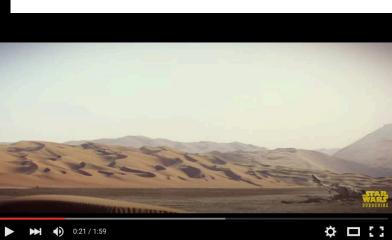
MML Language



Variants and configurations



Generator
~ composition of
video sequences



**video
variants**



Generator
~ composition of
video sequences

**video
variants**

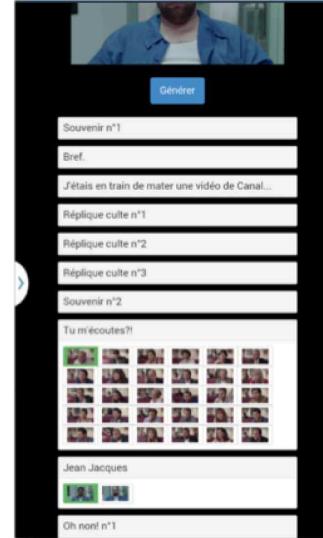
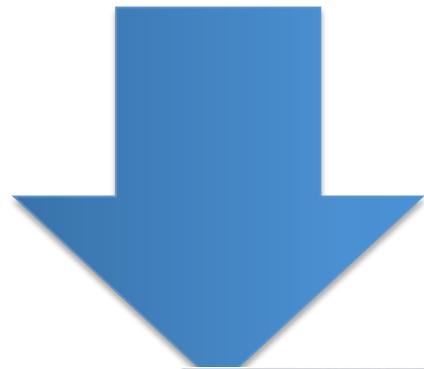




foo1.videogen ✎

```
mandatory videoseq v1 "https://www.youtube.com/watch?v=PjNi1uYhV5w"
optional videoseq v2 "v2/folder/v2.mp4"
alternatives v3 {
    videoseq v31 "v3/seq1.mp4"
    videoseq v32 "v3/seq1.mp4"
    videoseq v33 "v3/seq1.mp4"
}

alternatives v4 {
    videoseq v41 "v4/seq1.mp4"
    videoseq v42 "v4/seq1.mp4"
}
mandatory videoseq v5 "https://www.youtube.com/watch?v=ezKx-S0LiNQ"
```



- ## Website/online
- Random generation
 - Configurator
 - Game
 - ...

foo1.videogen ✘

```
mandatory videoseq v1 "https://www.youtube.com/watch?v=PjNi1uYhV5w"
optional videoseq v2 "v2folder/v2.mp4"
alternatives v3 {
    videoseq v31 "v3/seq1.mp4"
    videoseq v32 "v3/seq1.mp4"
    videoseq v33 "v3/seq1.mp4"
}
alternatives v4 {
    videoseq v41 "v4/seq1.mp4"
    videoseq v42 "v4/seq1.mp4"
}
mandatory videoseq v5 "https://www.youtube.com/watch?v=ezKx-S0LiNQ"
```

#1 How to design,
create, and support
dedicated languages
(DSLs)?

#2 How to transform
models/programs?



#3 How to manage
variability/variants?

#4 How do frameworks
internally work?

Plan

- Software product lines, configurable systems, and generators in the real-world
- From copy-and-paste to metamodeling
- The Jhipster case
 - An example of a real-world highly configurable system: how it is implemented and how we can model/test JHipster

Contract

- The idea of software product lines and variability
- Variability modeling
- Case studies

Software Product Line and Variability Engineering

karoma - Poderosa

File Edit Console Tools Window Plug-in Help

Line feed CR Encoding iso-8859-1 generic

karoma karoma

.config - Linux Kernel v2.6.33.3 Configuration

Processor type and features

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <V> includes, <X> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module capable <> module capable

[] Tickless System (Dynamic Ticks)

[*] High Resolution Timer Support

[] Symmetric multi-processing support

[] Support for extended (non-PC) x86 platforms

[] Single-depth WCHAN output

[] Paravirtualized guest support --->

[] Memtest

Processor family (Generic-x86-64) --->

Preemption Model (No Forced Preemption (Server)) --->

[] Reroute for broken boot IRQs (NEW)

[] Machine Check / overheating reporting

[] Dell laptop support

[] /dev/cpu/microcode - microcode support

[] /dev/cpu/*/msr - Model-specific register support

[] /dev/cpu/*/cpuid - CPU information support

Memory model (Sparse Memory) --->

[*] Sparse Memory virtual memmap (NEW)

[] Allow for memory hot-add (NEW)

[] Enable KSM for page merging

(4096) Low address space to protect from user allocation

[] Check for low memory corruption

[] Reserve low 64K of RAM on AMI/Phoenix BIOSen

-*- MTRR (Memory Type Range Register) support

[] MTRR cleanup support

[] Enable seccomp to safely compute untrusted bytecode

[] Enable -fstack-protector buffer overflow detection (EXPERIMENTAL)

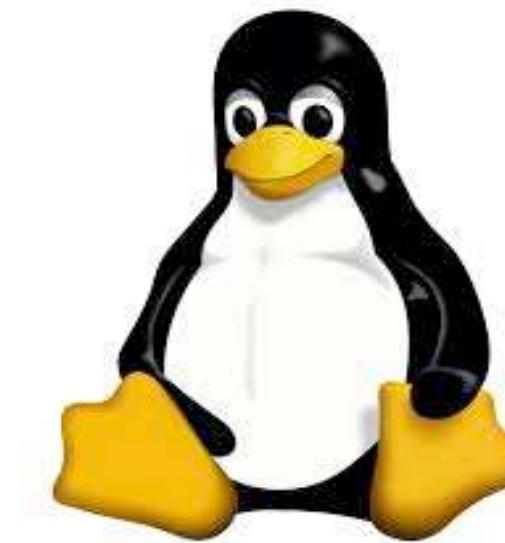
Timer frequency (250 Hz) --->

[] kexec system call

q(+)

<Select> <Exit> <Help>

Kernel Linux





Linux everywhere since highly configurable

```
config X86_X2APIC
    bool "Support x2apic"
    depends on X86_LOCAL_APIC && X86_64 && (IRQ_REMAP || HYPERVISOR_GUEST)
    ---help---
        This enables x2apic support on CPUs that have this feature.
```

This allows 32-bit apic IDs (so it can support very large systems),

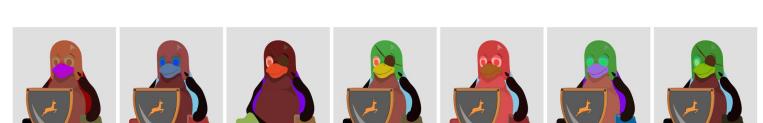
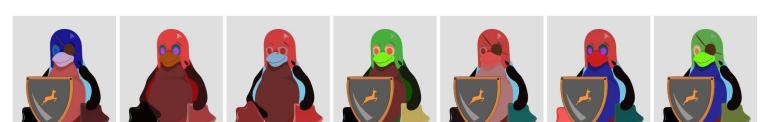
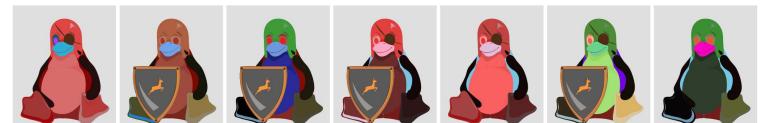
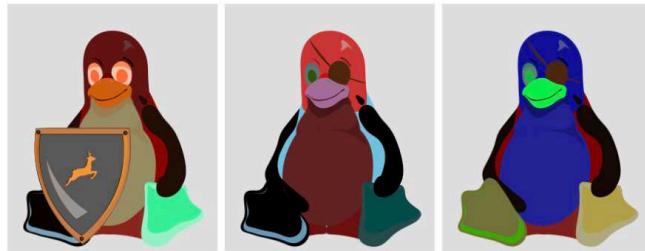
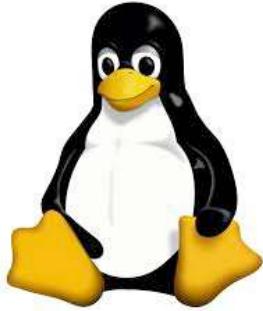
```
config IOSF_MBI
    tristate "Intel SoC IOSF Sideband support for SoC platforms"
    depends on PCI
    ---help---
        This option enables sideband register access support for Intel SoC
        platforms. On these platforms the IOSF sideband is used in lieu of
        MSR's for some register accesses, mostly but not limited to thermal
        and power. Drivers may query the availability of this device to
        determine if they need the sideband in order to work on these
        platforms. The sideband is available on the following SoC products.
```

```
#
# Processor type and features
#
# CONFIG_ZONE_DMA is not set
# CONFIG_SMP is not set
# CONFIG_X86_FEATURE_NAMES is not set
# CONFIG_X86_FAST_FEATURE_TESTS is not set
CONFIG_X86_X2APIC=y
CONFIG_X86_MPPARSE=y
CONFIG_GOLDFISH=y
# CONFIG_INTEL_RDT_A is not set
# CONFIG_X86_EXTENDED_PLATFORM is not set
CONFIG_IOSF_MBI=m
CONFIG_IOSF_MBI_DEBUG=y
CONFIG_X86_SUPPORTS_MEMORY_FAILURE=y
# CONFIG_SCHED OMIT_FRAME_POINTER is not set
```

Kconfig files/doc

.config

<https://github.com/diverse-project/tuxart>



You Retweeted



Thomas Thüm

@ThomasThuem

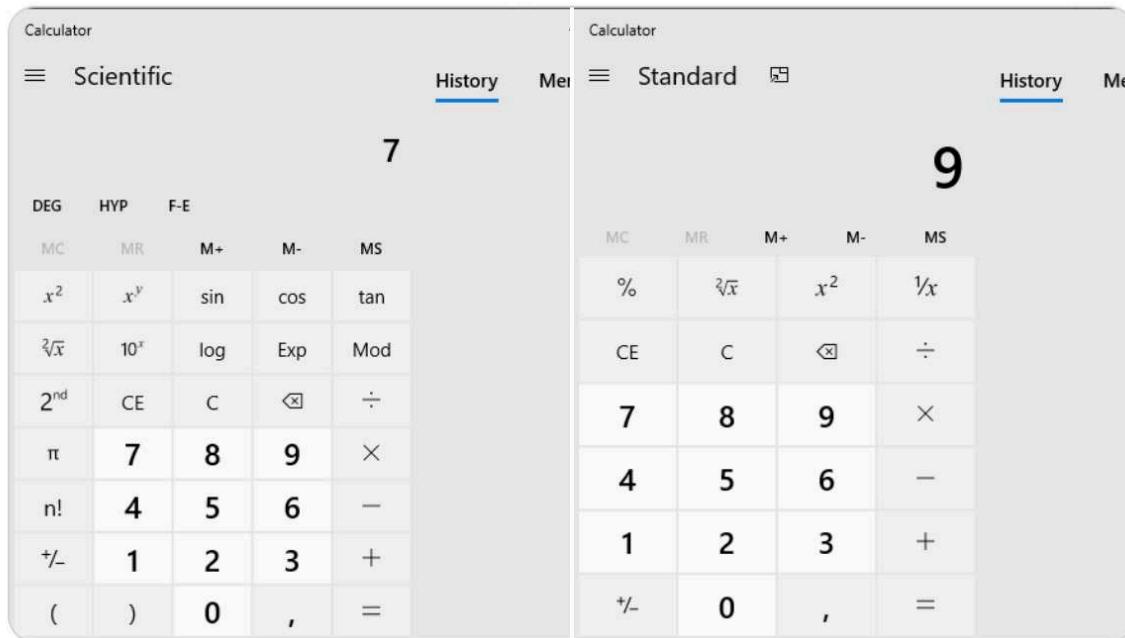
▼

How much is $1 + 2 * 3$?

According to [@Windows 10](#), the result depends on the mode that you have chosen!!!!

The Calculator app gives the correct result in Scientific mode and a wrong result with the default mode called Standard.

I bet this caused thousands of wrong calculations!





```
macher-wifi:getting-started macher1$ yo jhipster
```

I'm all done. Running `npm install & bower install` for you to install the required dependencies.

THE PAPER CLOTH
OF THE
WALLS,
THE
CLOTH
OF THE
WALLS,
THE
CLOTH
OF THE
WALLS,

Welcome to the JHipster Generator v2.17.0

```
? (1/15) What is the base name of your application? jhipster
? (2/15) What is your default Java package name? com.mycompany.myapp
? (3/15) Do you want to use Java 8? Yes (use Java 8)
? (4/15) Which *type* of authentication would you like to use? (Use arrow keys)
❯ HTTP Session Authentication (stateful, default Spring Security mechanism)
  OAuth2 Authentication (stateless, with an OAuth2 server implementation)
  Token-based authentication (stateless, with a token)
```

**Starter****Home Premium Upgrade****Professional Upgrade****Ultimate Upgrade**

\$119.99*

[Buy](#)

\$199.99*

[Buy](#)

\$219.99*

[Buy](#)

Communication

Bluetooth support	✓	✓	✓	✓
Join a homegroup	✓	✓	✓	✓
Internet Explorer 8	✓	✓	✓	✓
View Available Networks	✓	✓	✓	✓
Windows Connect Now (WCN)	✓	✓	✓	✓
Create a homegroup		✓	✓	✓
Location and other sensors support		✓	✓	✓
Support for joining domains			✓	✓

Entertainment

DirectX 11	✓	✓	✓	✓
Gadgets	✓	✓	✓	✓
Games Explorer	✓	✓	✓	✓
Play To	✓	✓	✓	✓
Windows Media Player 12	✓	✓	✓	✓
Create and play DVDs	✓	✓	✓	✓
Internet TV	✓	✓	✓	✓



(a) Variant #1 of video sequence



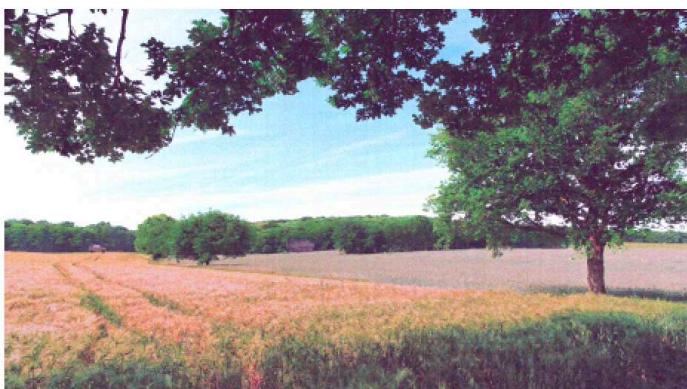
(b) Variant #2 of video sequence



(c) Variant #3 of video sequence



(d) Variant #4 of video sequence



(e) Variant #5 of video sequence



(f) Variant #6 of video sequence

Figure 1: Six variants of video sequences synthesized with ViViD

/* [Customize body] */

//Set the outside length of your pencil box.
length=190;//[70:400]

//Set the outside depth of your pencil box.
depth=70;//[50:400]

//Set the total height of your pencil box. The top of the box is set at 15mm.

//Extra height is added to the body section.

height=40;//[40:150]

//Choose divider orientation. Long is for the X direction.

long = 1;//[0,1,2]

//Short is for the Y direction.

short = 2;//[0,1,2,3]

//When you have 2 long dividers,

// picking yes here will put short dividers in the center section.

center = 0;//[1:Yes,0>No]

1

Lid inside settings

Lid inside content

Lid outside

Customize body

Design key

Customize ruler

Printer platform se

Length Set the outside length of your pencil box. 190

Depth Set the outside depth of your pencil box. 70

Height Set the total height of your pencil box. The top of the box is set at 15mm. Extra height is added to the body section. 40

Long Choose divider orientation. Long is for the X direction.

 1

Short Short is for the Y direction.

 2

Center When you have 2 long dividers, picking yes here will put short dividers in the center section.

Customizable Battery Case
by water, published Mar 5, 2013



Thing Info

Instructions

Thing Files

20

8

473

366

20865 2444

Found in Containers

Report Thing as Inappropriate

Makes

view more >

Description

A customizable battery case to hold batteries while traveling. Configurable for the number of batteries and type (as long as they're cylindrical). This is a updated version of the customizable battery carrier ([thingiverse.com/thing:51376](#)), re-designed to work without magnets as requested by GregFisk25.



Bref

bref.
CANAL à 30 ans.

ETAPE 1 : DONNE TON PRENOM

MATHIEU

→ OK

Online Generator

← → C bref30ans.canalplus.fr/#c

ETAPE 2 : CHOISIS 3 BONS SOUVENIRS



Variant



Quizz Time

Give three examples of software product lines (also called configurable systems or variability-intensive systems)

A large, intricate 3D white maze is set against a light gray background. The maze is composed of many interconnected paths and dead ends, creating a sense of complexity and depth. The perspective is from above, looking down into the various levels and recesses of the labyrinth.

Variability = Complexity

33 features



a unique variant for every
person on this planet

320^{optional, independent}
features

more variants than estimated
atoms in the universe



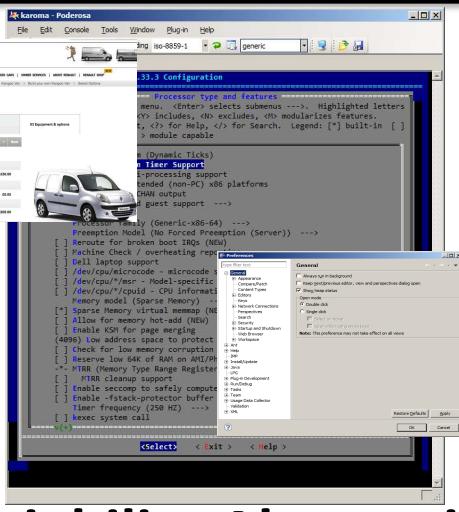
2000 features

16000
features



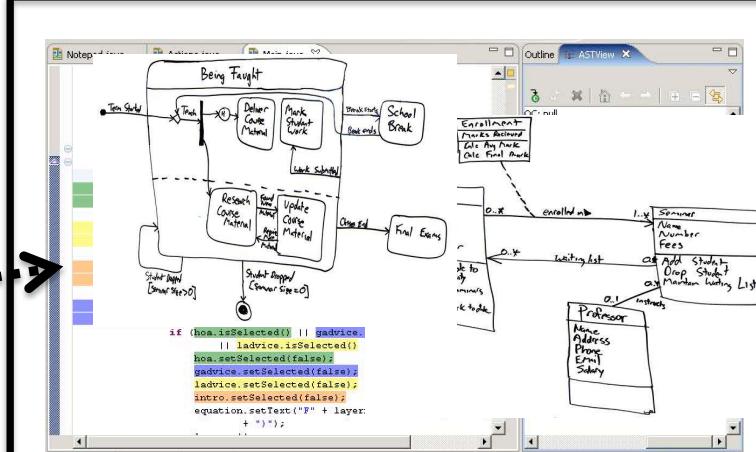


Linux Kernel
 $\approx 10^{80}$ configurations
 **$\approx 10^{80}$ is the estimated number of atoms
in the universe**
 **$\approx 10^{40}$ is the estimated number of
possible chess positions**



Variability Abstraction Model (VAM)

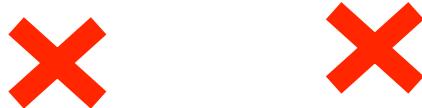
Variability
Realization
Model
(VRM)



Domain Artefacts
(e.g., models)



Configuration
(resolution model)



Software Generator
(derivation engine)



[generator-jhipster / app / templates / src / main / java / package / config / _DatabaseConfiguration.java](#) **jdubois** 2 days ago Use Spring Boot's configuration meta-data9 contributors 

184 lines (165 sloc) | 9.69 KB

[Raw](#) [Blame](#) [History](#)   

```
1 package <%=packageName%>.config;
2 <% if (databaseType == 'sql') { %>
3 import <%=packageName%>.config.liquibase.AsyncSpringLiquibase;
4 import com.codahale.metrics.MetricRegistry;
5 import com.fasterxml.jackson.datatype.hibernate4.Hibernate4Module;
6 import com.zaxxer.hikari.HikariConfig;
7 import com.zaxxer.hikari.HikariDataSource;
8 import liquibase.integration.spring.SpringLiquibase;<% } %><% if (databaseType == 'mongodb' && authenticationType == 'oauth2') { %>
9 import <%=packageName%>.config.oauth2.OAuth2AuthenticationReadConverter;<% } %><% if (databaseType == 'mongodb') { %>
10 import com.mongodb.Mongo;
11 import org.mongeez.Mongeez;<% } %>
12 import org.slf4j.Logger;
13 import org.slf4j.LoggerFactory;<% if (databaseType == 'sql') { %><% if (hibernateCache == 'hazelcast') { %>
14 import org.springframework.cache.CacheManager;<% } %>
15 import org.springframework.beans.factory.annotation.Autowired;
16 import org.springframework.boot.autoconfigure.condition.ConditionalOnExpression;<% } %><% if (databaseType == 'mongodb') { %>
17 import org.springframework.boot.autoconfigure.mongo.MongoAutoConfiguration;
18 import org.springframework.boot.autoconfigure.mongo.MongoProperties;<% } %><% if (databaseType == 'sql') { %>
19 import org.springframework.boot.autoconfigure.jdbc.DataSourceProperties;
20 import org.springframework.boot.autoconfigure.liquibase.LiquibaseProperties;
21 import org.springframework.context.ApplicationContextException;<% } %>
22 import org.springframework.context.annotation.Bean;
23 import org.springframework.context.annotation.Configuration;
24 import org.springframework.context.annotation.Profile;<% if (databaseType == 'mongodb') { %>
25 import org.springframework.context.annotation.Import;<% } %><% if (databaseType == 'sql') { %>
26 import org.springframework.core.env.Environment;<% } %><% if (databaseType == 'mongodb' && authenticationType == 'oauth2') { %>
27 import org.springframework.core.convert.converter.Converter;<% } %><% if (databaseType == 'mongodb') { %>
28 import org.springframework.core.io.ClassPathResource;<% } %><% if (searchEngine == 'elasticsearch') { %>
29 import org.springframework.data.elasticsearch.repository.config.EnableElasticsearchRepositories;<% } %><% if (databaseType == 'mon
30 import org.springframework.data.mongodb.config.AbstractMongoConfiguration;
31 import org.springframework.data.mongodb.config.EnableMongoAuditing;<% } %><% if (databaseType == 'mongodb' && authenticationType =
32 import org.springframework.data.mongodb.core.convert.CustomConversions;<% } %><% if (databaseType == 'mongodb') { %>
33 import org.springframework.data.mongodb.core.mapping.event.ValidatingMongoEventListener;
34 import org.springframework.data.mongodb.repository.config.EnableMongoRepositories;
35 import org.springframework.validation.beanvalidation.LocalValidatorFactoryBean;<% } %><% if (databaseType == 'sql') { %>
```

macher-wifi:getting-started macher1\$ yo jhipster

I'm all done. Running `npm install & bower install` for you to install the required dependencies.

JHIPSTER STARER IF YOU DON'T KNOW

Welcome to the JHipster Generator v2.17.0

- ? (1/15) What is the base name of your application? `jhipster`
- ? (2/15) What is your default Java package name? `com.mycompany.myapp`
- ? (3/15) Do you want to use Java 8? Yes (use Java 8)
- ? (4/15) Which type of authentication would you like to use? (Use arrow keys)
 - > `HTTP Session Authentication` (stateful, default Spring Security mechanism)
 - `OAuth2 Authentication` (stateless, with an OAuth2 server implementation)
 - `Token-based authentication` (stateless, with a token)

Variability Model

mapping

Branch: master

generator-jhipster / app / templates / src / main / java / package / config / _DatabaseConfiguration.java

l dlbous 2 days ago Use Spring Boot's configuration meta-data

9 contributors

Raw Blame History

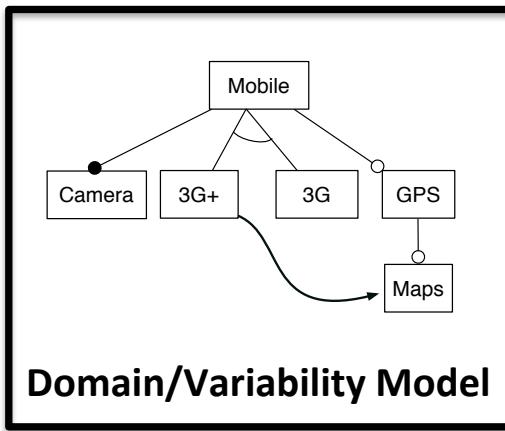
```
1 package com.mycompany.myapp.config;
2 
3 import com.codahale.metrics.MetricRegistry;
4 import com.fasterxml.jackson.databind.ObjectMapper;
5 import com.zaxxer.hikari.HikariConfig;
6 import com.zaxxer.hikari.HikariDataSource;
7 import org.hibernate.cache.CacheManager;
8 import org.hibernate.cache.hazelcast.HazelcastCacheManager;
9 import org.hibernate.cache.jcache.JCacheFactory;
10 import org.hibernate.cache.spi.CacheConcurrencyStrategy;
11 import org.hibernate.cache.spi.TimestampingType;
12 import org.hibernate.cache.spi.TimestampingType;
13 import org.hibernate.cache.spi.TimestampingType;
14 import org.hibernate.cache.spi.TimestampingType;
15 import org.hibernate.cache.spi.TimestampingType;
16 import org.hibernate.cache.spi.TimestampingType;
17 import org.hibernate.cache.spi.TimestampingType;
18 import org.hibernate.cache.spi.TimestampingType;
19 import org.hibernate.cache.spi.TimestampingType;
20 import org.hibernate.cache.spi.TimestampingType;
21 import org.hibernate.cache.spi.TimestampingType;
22 import org.hibernate.cache.spi.TimestampingType;
23 import org.hibernate.cache.spi.TimestampingType;
24 import org.hibernate.cache.spi.TimestampingType;
25 import org.hibernate.cache.spi.TimestampingType;
26 import org.hibernate.cache.spi.TimestampingType;
27 import org.hibernate.cache.spi.TimestampingType;
28 import org.hibernate.cache.spi.TimestampingType;
29 import org.hibernate.cache.spi.TimestampingType;
30 import org.hibernate.cache.spi.TimestampingType;
31 import org.hibernate.cache.spi.TimestampingType;
32 import org.hibernate.cache.spi.TimestampingType;
33 import org.hibernate.cache.spi.TimestampingType;
34 import org.hibernate.cache.spi.TimestampingType;
35 import org.hibernate.cache.spi.TimestampingType;
```

Base Artefacts



Software Generator (derivation engine)

Domain Engineering

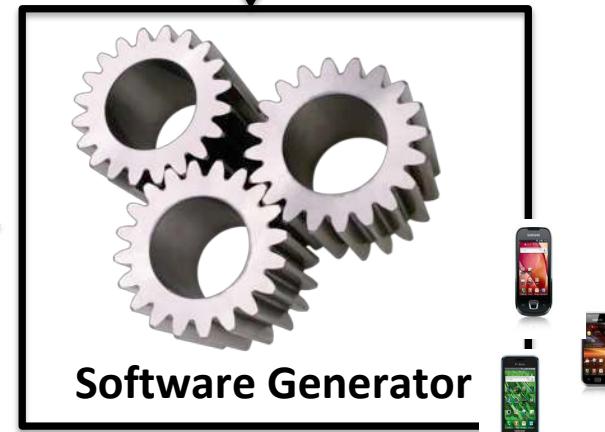
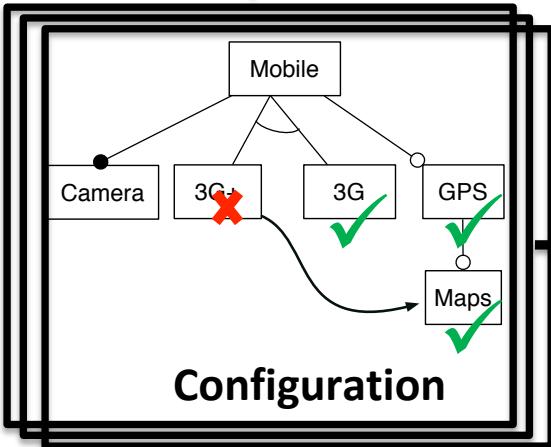


A screenshot of a software interface showing code in Notepad.java and an AST View panel. The code includes logic for selecting devices based on user input.

```
Notepad.java
Actions.java
Main.java 23
    output.setText(t.toString());
    program.setText(t.eval(""));
    equation.setText(base);
    updateQuarkPanel();
}
apply.addChangeListener(ActionListener) {
    public void actionPerformed(ActionEvent e) {
        if (device.isSelected()) {
            t = t.apply(new hoan("b" + layerno));
        }
        if (advise.isSelected()) {
            t = t.apply(new advise("a" + layerno));
        }
        if (intro.isSelected()) {
            t = t.apply(new intro("i" + layerno));
        }
        if (gadvice.isSelected()) {
            t = t.apply(new gadvice("g" + layerno));
        }
        if (hoan.isSelected() || gadvice.isSelected() || ladvice.isSelected() || intro.isSelected()) {
            hoan.setSelected(false);
            gadvice.setSelected(false);
            ladvice.setSelected(false);
            intro.setSelected(false);
            equation.setText("f" + layerno + "(" + equation.getText() + ")");
        }
    }
}
```

AST View panel showing the parse tree structure of the code.

Application Engineering



« the investments required to develop the reusable artifacts during **domain engineering**, are outweighed by the benefits of deriving the individual products during **application engineering** »

« variability »

Is it really new?

Command Line Options

```
x264 --quiet  
      --no-progress  
      --no-asm  
      --rc-lookahead 60  
      --ref 9  
      -o trailer_480p24.x264  
      trailer_2k_480p24.y4m
```

Parameter -i in grep

```
1 int match_icase;
2
3 int main (int argc, char **argv)
4 {
5     [...]
6     while ((opt = get_nondigit_option (argc, argv, &default_color))
7         switch (opt)
8         {
9             [...]
10            case 'i':
11                match_icase = 1;
12                break;
13            }
14        }
15
16
17 static const char *
18 print_line_middle (const char *beg, const char *lim,
19                     const char *line_color, const char *match_color)
20 {
21     [...]
22     if (match_icase)
23     {
24         ibeg = buf = (char *) xmalloc(i);
25         while (--i >= 0)
26             buf[i] = tolower(beg[i]);
27     }
}
```

Global configuration

```
class Config {  
    public static boolean isLogging = false;  
    public static boolean isWindows = false;  
    public static boolean isLinux = true;  
}  
class Main {  
    public void foo() {  
        if (isLogging)  
            log(„running foo()“);  
        if (isWindows)  
            callWindowsMethod();  
        else if (isLinux)  
            callLinuxMethod();  
        else  
            throw RuntimeException();  
    }  
}
```

Configuration

httpd.conf -- win32 Apache Building a Web Server, for Windows

```
Listen 80
ServerRoot "/www/Apache2"
DocumentRoot "/www/webroot"
```

```
ServerName localhost:80
ServerAdmin admin@localhost
```

```
ServerSignature On
ServerTokens Full
```

```
DefaultType text/plain
AddDefaultCharset ISO-8859-1
```

```
UseCanonicalName Off
```

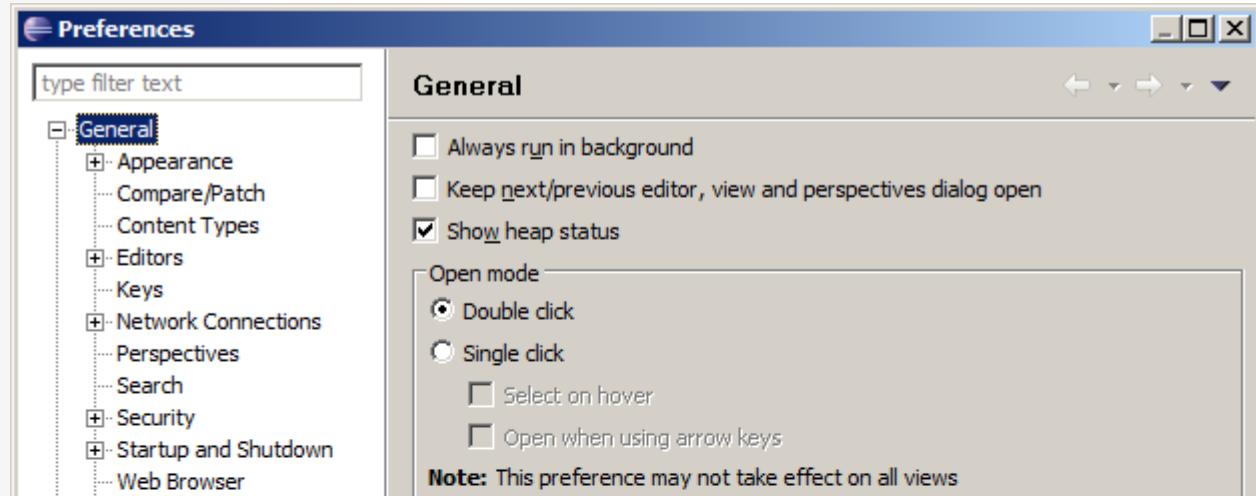
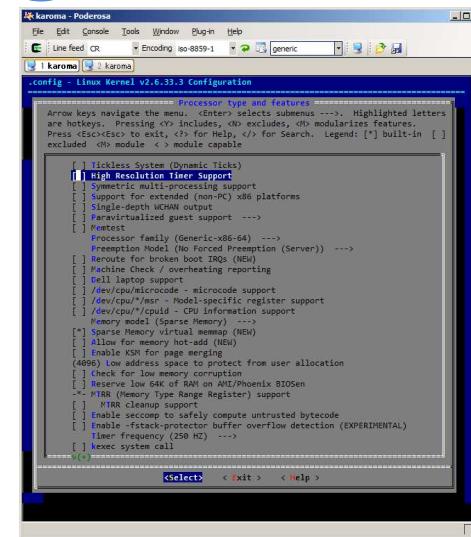
```
HostnameLookups Off
```

```
ErrorLog logs/error.log
LogLevel error
```

```
PidFile logs/httpd.pid
```

```
Timeout 300
```

```
KeepAlive On
MaxKeepAliveRequests 100
```



Conditional compilation

#ifdef (Berkeley DB)

```
static int __rep_queue_filedone(dbenv, rep, rfp)
    DB_ENV *dbenv;
    REP *rep;
    __rep_fileinfo_args *rfp; {
#ifndef HAVE_QUEUE
    COMPQUIET(rep, NULL);
    COMPQUIET(rfp, NULL);
    return (__db_no_queue_am(dbenv));
#else
    db_pgno_t first, last;
    u_int32_t flags;
    int empty, ret, t_ret;
#endif
#ifdef DIAGNOSTIC
    DB_MSGBUF mb;
#endif
    // over 100 lines of additional code
}
#endif
```

Intentional Code Cloning

~ Copy & Paste

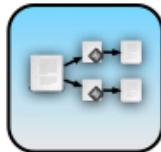
Code Cloning (example, Linux driver)

cyberstorm.c

```
....  
static void dma_dump_state(struct NCR_ESP *esp)  
{  
    ESPLOG("esp%d: dma -- cond_reg<%02x>\n",  
           esp->esp_id, ((struct cyber_dma_registers *)  
                           (esp->dregs))->cond_reg);  
    ESPLOG("intreq:<%04x>, intena:<%04x>\n",  
           custom.intreq, custom.intenar));  
}  
  
static void dma_init_read(struct NCR_ESP *esp, __u32 addr, int  
length)  
{  
    struct cyber_dma_registers *dregs =  
        (struct cyber_dma_registers *) esp->dregs;  
  
    cache_clear(addr, length);  
  
    addr &= ~(1);  
    dregs->dma_addr0 = (addr >> 24) & 0xff;  
    dregs->dma_addr1 = (addr >> 16) & 0xff;  
    dregs->dma_addr2 = (addr >> 8) & 0xff;  
    dregs->dma_addr3 = (addr ) & 0xff;  
    ctrl_data &= ~(CYBER_DMA_WRITE);  
}.....
```

cyberstormll.c

```
....  
static void dma_dump_state(struct NCR_ESP *esp)  
{  
    ESPLOG("esp%d: dma -- cond_reg<%02x>\n",  
           esp->esp_id, ((struct cyberll_dma_registers *)  
                           (esp->dregs))->cond_reg));  
    ESPLOG("intreq:<%04x>, intena:<%04x>\n",  
           custom.intreq, custom.intenar));  
}  
  
static void dma_init_read(struct NCR_ESP *esp, __u32 addr, int  
length)  
{  
    struct cyberll_dma_registers *dregs =  
        (struct cyberll_dma_registers *) esp->dregs;  
  
    cache_clear(addr, length);  
  
    addr &= ~(1);  
    dregs->dma_addr0 = (addr >> 24) & 0xff;  
    dregs->dma_addr1 = (addr >> 16) & 0xff;  
    dregs->dma_addr2 = (addr >> 8) & 0xff;  
    dregs->dma_addr3 = (addr ) & 0xff;  
}  
.....
```

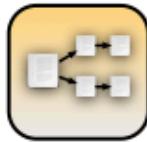


Replicate & Specialize

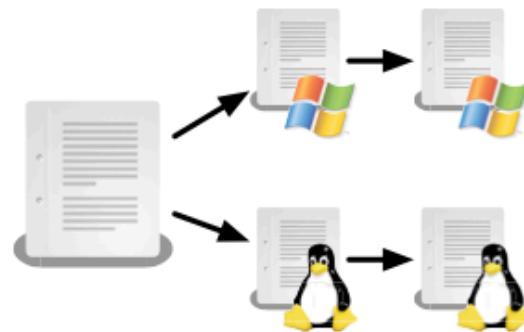


Clone to reuse and adapt existing solutions

- + Less effort needed
- Long-term cost outweighs short-term benefit
- ~ Cost of refactoring rises over time

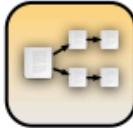


Platform Variations



**Clone existing code and fix
low level platform interaction**

- + Avoid complexity of virtualization layer
- Hard to propagate bug fixes
- ~ Ensure consistent behavior of all clones



Hardware Variations



Clone existing driver

- + No risk of changing existing driver
- Code growth
- ~ Dead code can creep into system

Inheritance (OOP)

Base Class encapsulate commonalities

Derive classes specialize peculiarities

Generic Programming

C++ template

```
template <typename T>
T max(T x, T y)
{
    return x < y ? y : x;
}
```

Generics in Java

```
public interface List<E> {
    void add(E x);
    Iterator<E> iterator();
}
public interface Iterator<E> {
    E next();
    boolean hasNext();
}
```

Design Patterns

Template Method

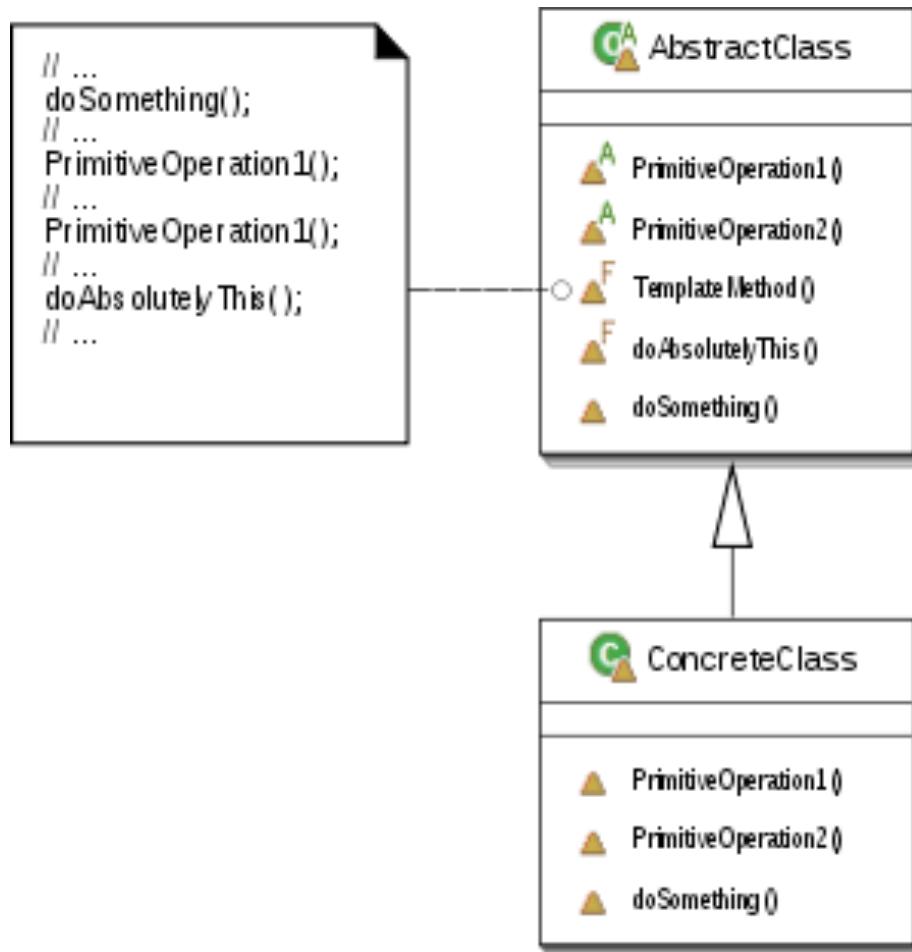
Factory

Strategy

Decorator

....

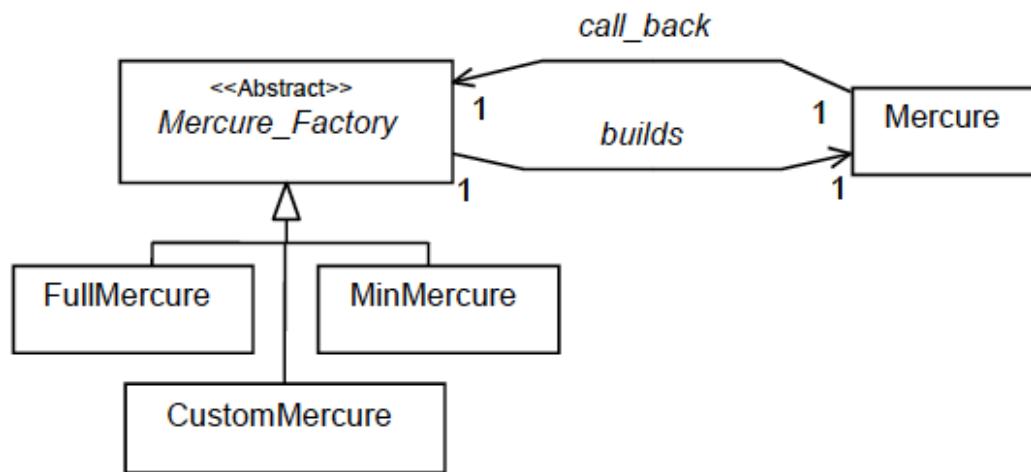
Template Method



The decision model

■ The Abstract Factory Design Pattern – [Gamma et al 95]

Mercure_Factory
new_gui() : GUI
new_language() : Language
new_network_manager() : Manager
new_netdriver() : Net Driver
new_engine() : Engine



CustomMercure
<<GUI1>> <<GUI2>> new_gui() : GUI
<<<Language2-1>> new_language() : Language
<<Manager1>> new_network_manager() : Manager
<<NetDriver1>> <<NetDriver2>> new_netdriver() : Net Driver
<<Engine1>> new_engine() : Engine

API Framework

Plugin-based systems

(Active) Annotations

can have parameters

Metamodeling and Domain-Specific Languages

httpd.conf -- win32 Apache

Building a Web Server, for Windows

```
Listen 80
ServerRoot "/www/Apache2"
DocumentRoot "/www/webroot"

ServerName localhost:80
ServerAdmin admin@localhost

ServerSignature On
ServerTokens Full

DefaultType text/plain
AddDefaultCharset ISO-8859-1

UseCanonicalName Off

HostnameLookups Off

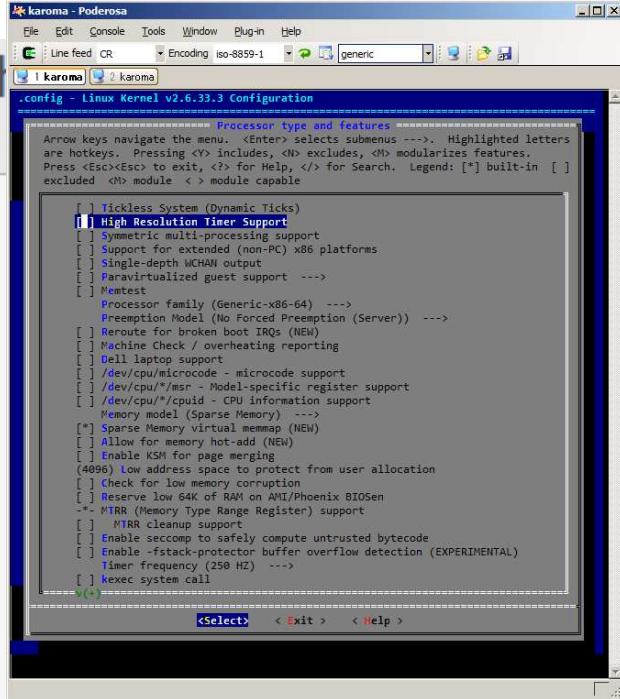
ErrorLog logs/error.log
LogLevel error

PidFile logs/httpd.pid

Timeout 300

KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

<IfModule mpm_winnt.c>
    ThreadsPerChild 250
    MaxRequestsPerChild 0
</IfModule>
```



A screenshot of the Renault Vans website. The top navigation bar includes links for CARS, VANS, ELECTRIC VEHICLES, RENAULT BUSINESS, USED CARS, OWNER SERVICES, ABOUT RENAULT, and RENAULT SHOP. Below this, a breadcrumb trail shows Renault UK > Renault Vans > New Kangoo Van Range > Kangoo Van > Build your own Kangoo Van > Selected Options. The main content area is titled 'NEW KANGOO VAN RANGE' and shows a configuration interface with three tabs: 01 Preferences, 02 Version, and 03 Equipment & options. Under 'OPTIONS', there are sections for 'COMFORT' (Central storage console & armrest between seats, £50.00), 'DRIVING' (Electric door mirrors, £0.00), and 'SAFETY & SECURITY' (ESC (Electronic Stability Control) with traction and understeer control, £200.00). To the right, there is a large image of a white Renault Kangoo van.

A screenshot of the Eclipse IDE. The central part shows a code editor with Java code for Notepad.java, Actions.java, and Main.java. The code includes various imports and methods like addActionListener and actionPerformed. The right side of the interface has several panes: Outline, Problems, Tasks, and AST View. The AST View pane shows the parse tree structure of the code. A note in the preferences pane says: 'Note: This preference is not yet supported by the Java UI.' The preferences pane also lists other options like 'Always run in background' and 'Keep next/previous editor, view and perspectives dialog open'.

A photograph of an old, rusted pickup truck. The truck is heavily damaged, with its front end crushed and the driver's side door missing. It is situated in a field of tall, dry grass and weeds, with a dense thicket of bushes in the background. The overall scene conveys a sense of abandonment and decay.

Unused flexibility



Illegal variant

Feature Models

A screenshot of the Karmic configuration tool interface. It displays a list of kernel configuration options under the heading "Processor Type and Features". The options include "High Resolution Timer Support", "Symmetric multi-processing support", "Single-depth MMU output", "Paravirtualized guest support", "Processor family (Generic<-x86-64)", "Preemption Model (No Forced Preemption (Server))", "Reserve first boot IRQ (NEW)", "Machine Check Exception reporting", "Bell laptop support", "Microcode support", "/dev/cpu/microcode", "Model-specific register support", "/dev/cpu/*cpu/* - CPU Information support", "Memory model (Sparse Memory) ...", "Allow memory hot-add (NEW)", "Enable KSM for page merging", "AppArmor security module (detect from user application)", "Check for low memory corruption", "Reserve low 64K of RAM on AMI/Phoenix BIOSen", "Support for Range Register support", "PTB cleanup support", "Enable seccomp to safely compute untrusted bytecode", "Enable ftrace-protector buffer overflow detection (EXPERIMENTAL)", "File freelist (256 KB) ...", and "/kexec system call". At the bottom, there are buttons for "<Select> < Exit > < Help >".

Variability Model

A screenshot of an IDE showing a Java code editor with several annotations overlaid. The code is part of a class named Notepad.java and includes imports for Actions.java and Main.java. Annotations are color-coded: green for 'hoa.isSelected()', yellow for 'ladvice.isSelected()', orange for 'intro.isSelected()', blue for 'gadvice.isSelected()', and red for 'hoa.setSelected()', 'ladvice.setSelected()', 'intro.setSelected()', and 'equation.setSelected()'. A large red arrow points from the "Variability Model" section to this code editor.

Modeling variability in main artifacts (e.g., source code)



Configuration



is crucial



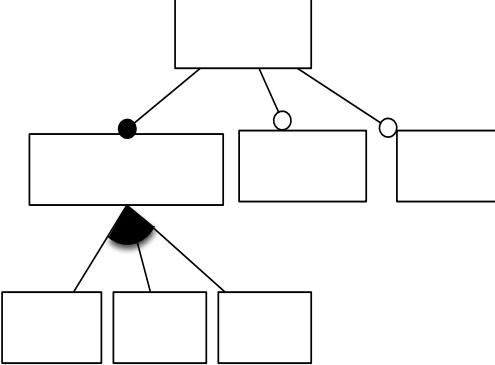
Unused flexibility





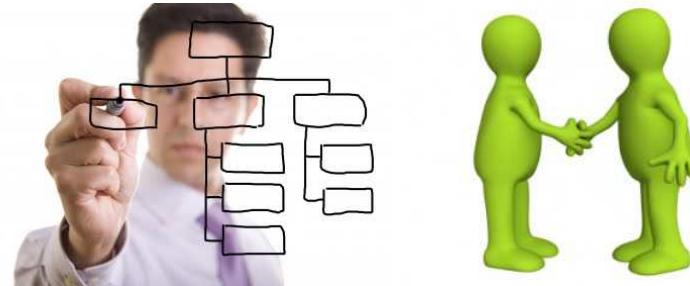
Illegal variant

Feature Model



not, and, or, implies

Communicative



Analytic



Generative





R8 Spyder 5.2 FSI quattro R-tronic

Prix total

171.216,00 EUR

Prix de base

170.490,00 EUR

Equipements optionnels

726,00 EUR

- ▶ Informations détaillées
- ▶ Entrez l'Audi Code
- ▶ Générer un PDF
- ▶ Nouvelle configuration

[+] Plein écran / Dimensions

▶ Fermer la capote

Habitatcl

Tableau de bord

Packs

Aucun pack n'est proposé pour ce modèle.

Couleurs

Blanc Ibis

Noir

Prix: 0,00 EUR



Couleurs métallisées à partir de 0,00 EUR



Couleurs à effet perlé à partir de 0,00 EUR



Couleurs personnalisées Audi exclusive



Couleur capote

Noir



Jantes

4 Jantes alu 5 BRANCHES ROTOR finition titane 8,5 x 19 à l'avant, 11 x 19 à l'arrière. Pneus 235/35 R19 à l'avant et 305 /30 R19 à l'arrière

Prix: 726,00 EUR

19" à partir de 0,00 EUR





R8 Spyder

5.2 FSI quattro R tronic

Prix total

185.899,35 EUR

Prix de base

170.490,00 EUR

Equipements optionnels

15.409,35 EUR

▶ Informations détaillées

▶ Entrez l'Audi Code

▶ Générer un PDF

▶ Nouvelle configuration

[+ Plein écran / Dimensions

[+ Vue extérieure

[+ Tableau de bord

▶ Packs d'équipements

▶ Extérieur

▶ Jantes & pneumatiques

▶ Intérieur

▶ Volants

▶ Sièges

Sécurité & technique

▶ Infotainment

▶ Châssis

▶ Freins

Systèmes d'assistance

▶ Autres

Régulateur de vitesse

320,65 EUR

Système d'aide au stationnement APS avant / arrière

931,70 EUR



Système d'aide au stationnement APS avant / arrière avec affichage dans l'écran MMI

1.373,35 EUR

Système d'aide au stationnement Advanced : APS avant et arrière et caméra arrière

1.790,80 EUR



Audi hill assist : assistance au démarrage en côte

Série

Réinitialiser la sélection

Attention:

Le prix peut varier en fonction du choix de moteur et des équipements.

Un aperç des équipements:

Mode expert

excludes





A5 Sportback 3.0 TDI quattro S tronic

Prix total

54.450,15 EUR

Prix de base

50.570,00 EUR

Equipements optionnels

3.890,15 EUR

▶ Informations détaillées

▶ Entrez l'Audi Code

▶ Nouvelle configuration

Vérification de votre sélection

Cet équipement nécessite un équipement complémentaire:

- GPS Plus avec disque dur 2.934,25 EUR

Voici les équipements complémentaires possibles:

- Ordinateur de bord en couleur avec programme efficiency 181,50 EUR
Remarque: uniquement sur les modèles avec système Start-Stop et uniquement disponible en combinaison avec l'autoradio Concert, l'autoradio Symphony ou un système de navigation
- Pack Intenso Plus 3.100,00 EUR
Sans appareil de navigation

[+] Plein écran / Dimensions

- ▶ Extérieur
- ▶ Jantes & pneumatiques
- ▶ Intérieur
- ▶ Volants
- ▶ Sièges
- ▶ Sécurité & technique
- Infotainment**

Attention:

Le prix peut varier en fonction du choix de moteur et des équipements.

Un aperç des équipements:

Mode expert

Réinitialiser la sélection

1 Modèle

2 Moteur

3 Extérieur

4 Intérieur

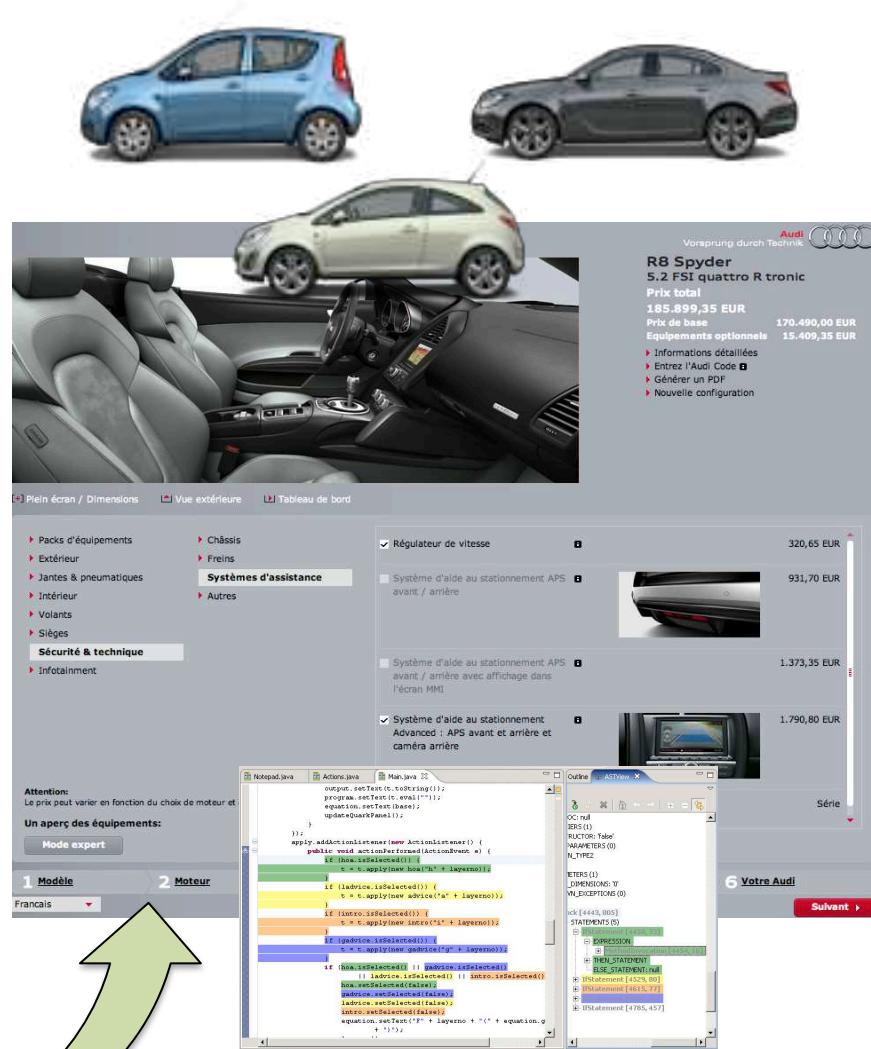
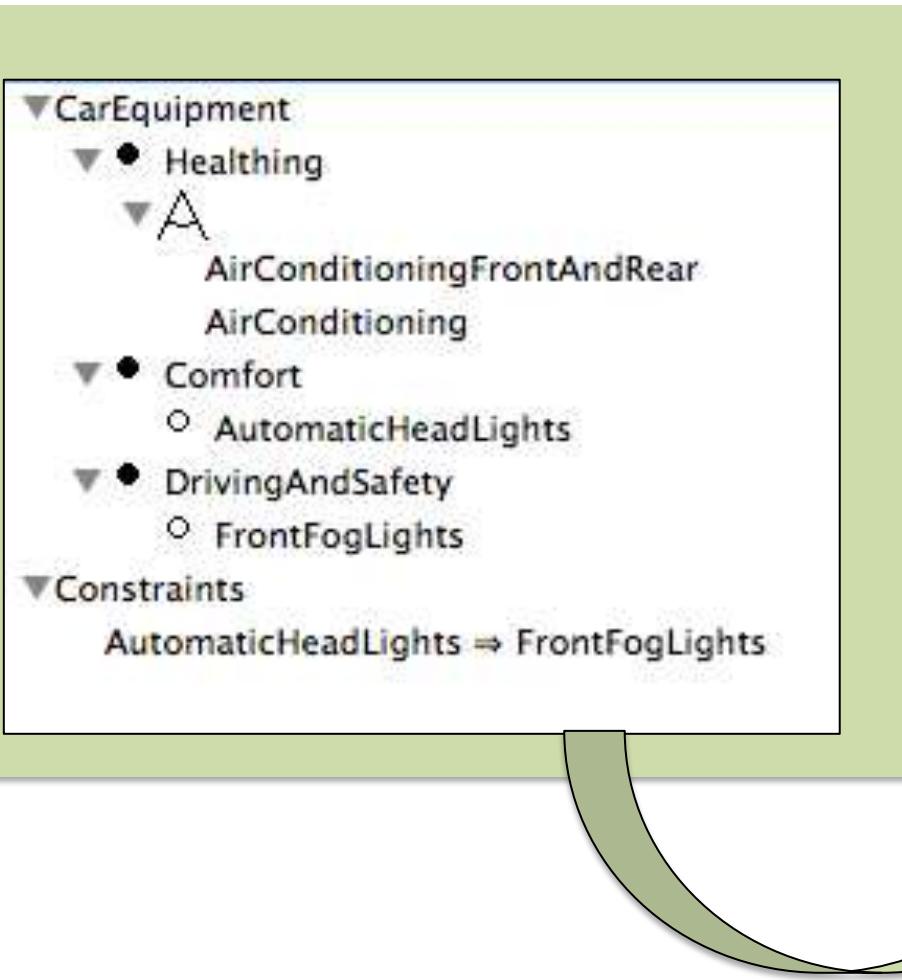
5 Option

6 Votre Audi

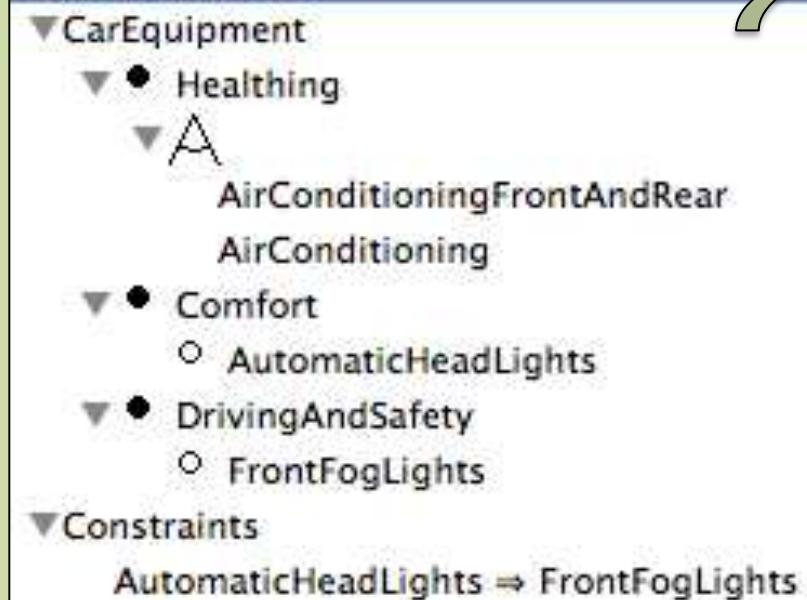
Français

Suivant ▶

Feature Models



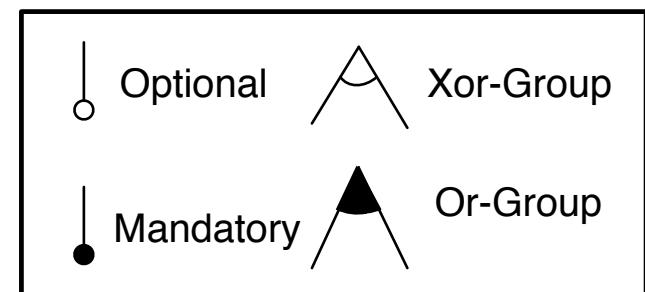
Feature Models (Background)

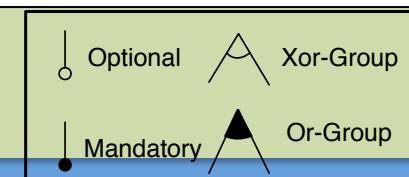
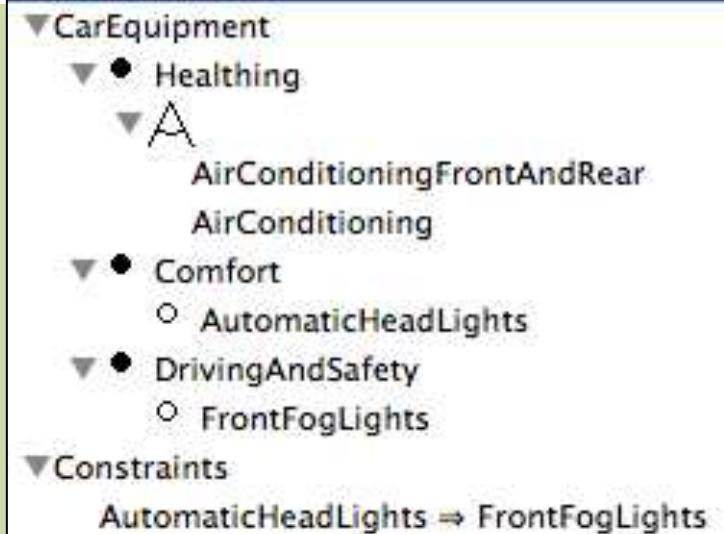
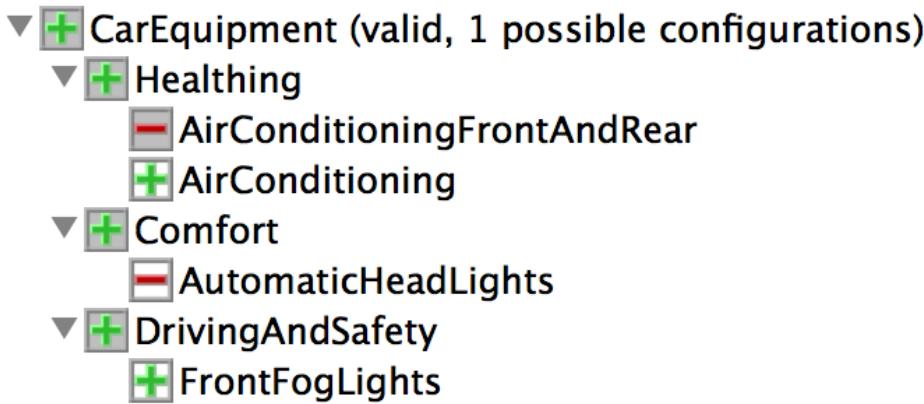


Hierarchy: rooted tree

Variability:

- mandatory,
- optional,
- Groups: exclusive or inclusive features
- Cross-tree constraints





Hierarchy + Variability

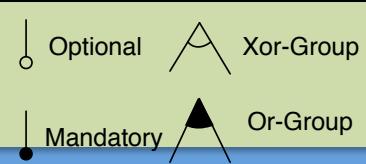
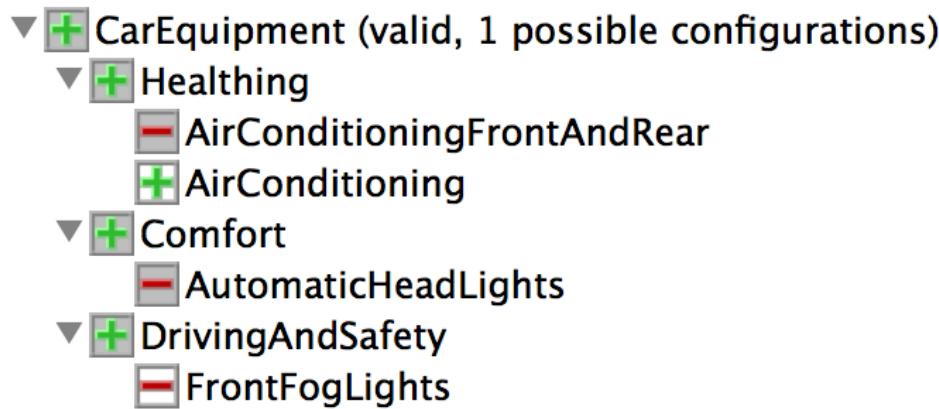
=

set of valid configurations

configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning, FrontFogLights}





Hierarchy + Variability

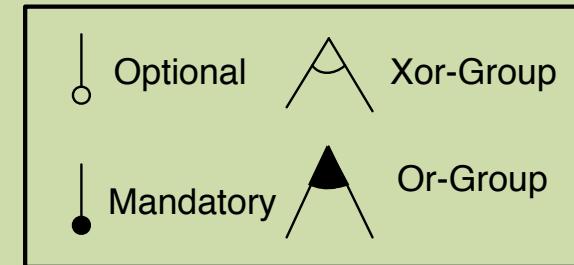
=

set of valid configurations

configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning}





Hierarchy + Variability

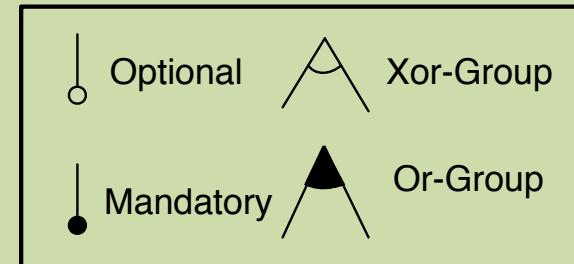
=

set of valid configurations

configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning, AutomaticHeadLights}



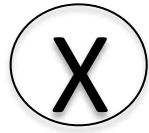


Hierarchy + Variability

=

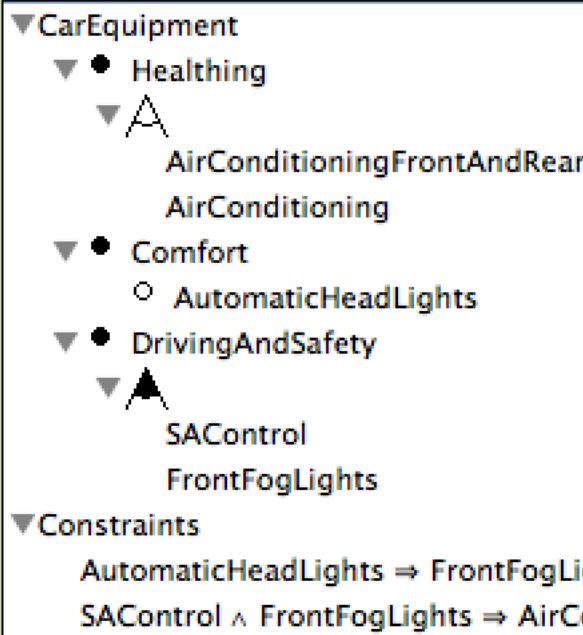
set of valid configurations

{CarEquipment, Comfort,
DrivingAndSafety,
Healthing}

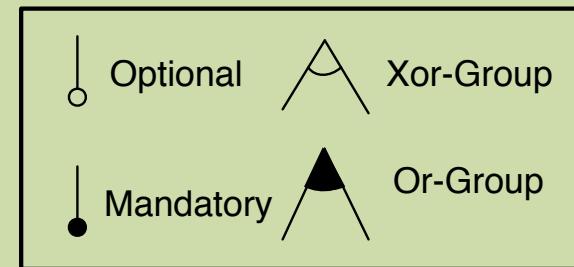


- {AirConditioning, FrontFogLights}
- {AutomaticHeadLights, AirConditioning, FrontFogLights}
- {AutomaticHeadLights, FrontFogLights, AirConditioningFrontAndRear}
- {AirConditioningFrontAndRear}
- {AirConditioning}
- {AirConditioningFrontAndRear, FrontFogLights}





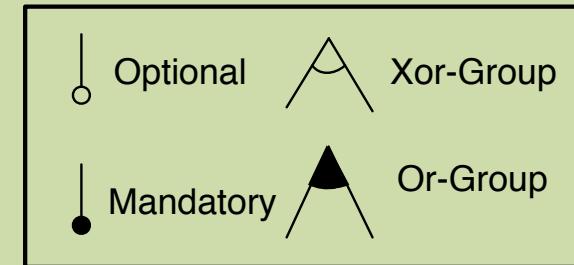
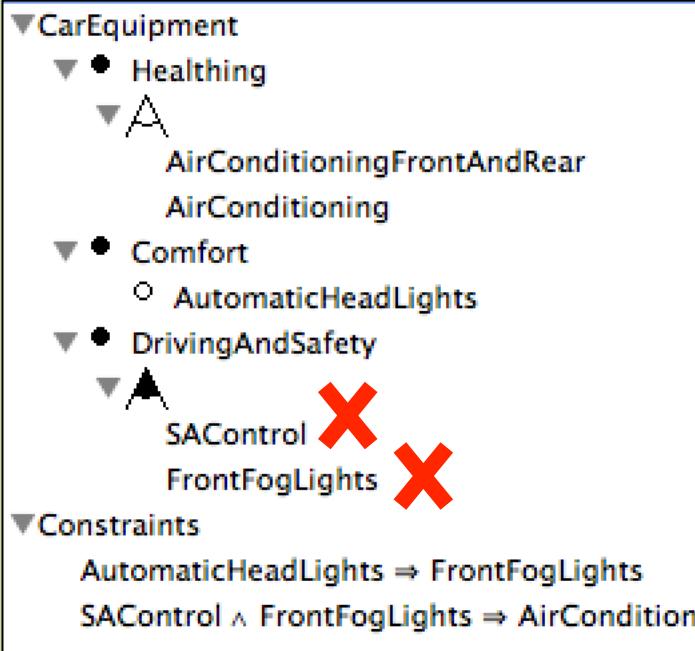
Boolean logic: \wedge , \vee , not, implies



Hierarchy + Variability
=

set of valid configurations





Hierarchy + Variability

=

set of valid configurations



Or-group: at least one!



Quizz Time

Enumerate all configurations of...

CarEquipment

• Healthing



AirConditioningFrontAndRear

AirConditioning

• Comfort

○ AutomaticHeadLights

• DrivingAndSafety



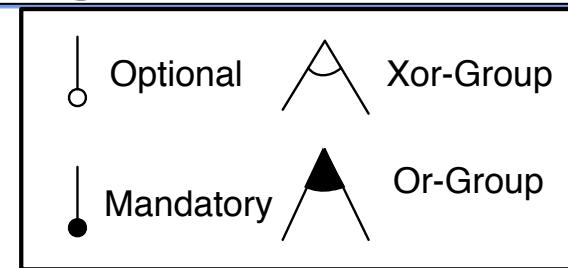
SAControl

FrontFogLights

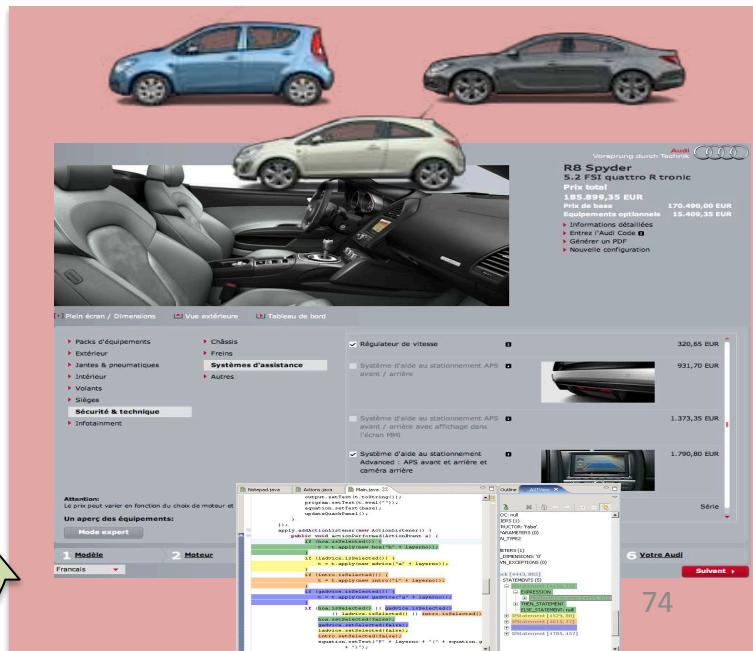
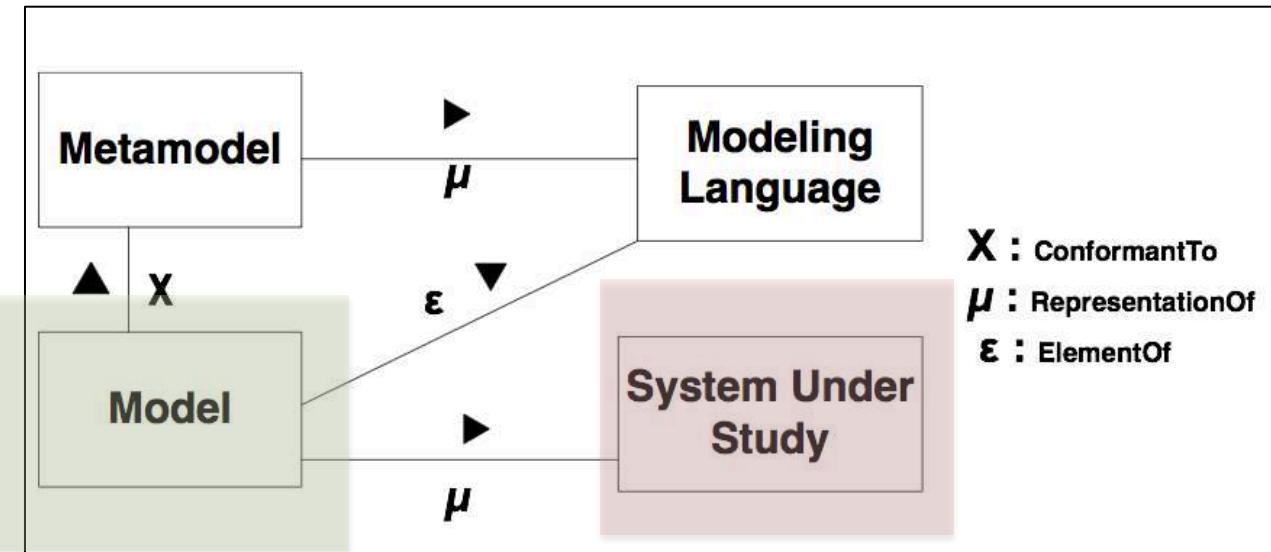
Constraints

AutomaticHeadLights \Rightarrow FrontFogLights

SAControl \wedge FrontFogLights \Rightarrow AirConditioningFrontAndRear



Feature Models

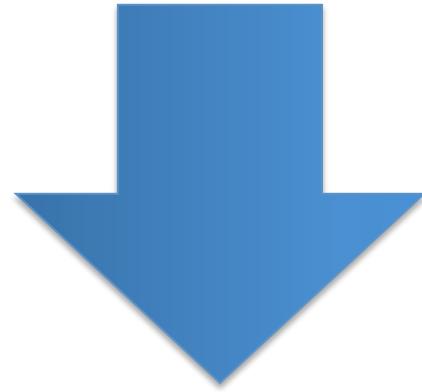




```
foo1.videogen ✘

mandatory videoseq v1 "https://www.youtube.com/watch?v=PjNi1uYhV5w"
optional videoseq v2 "v2folder/v2.mp4"
alternatives v3 {
    videoseq v31 "v3/seq1.mp4"
    videoseq v32 "v3/seq1.mp4"
    videoseq v33 "v3/seq1.mp4"
}

alternatives v4 {
    videoseq v41 "v4/seq1.mp4"
    videoseq v42 "v4/seq1.mp4"
}
mandatory videoseq v5 "https://www.youtube.com/watch?v=ezKx-S0LiNQ"
```



 FFmpeg

The FFmpeg logo, consisting of a green stylized 'F' followed by the word "FFmpeg".

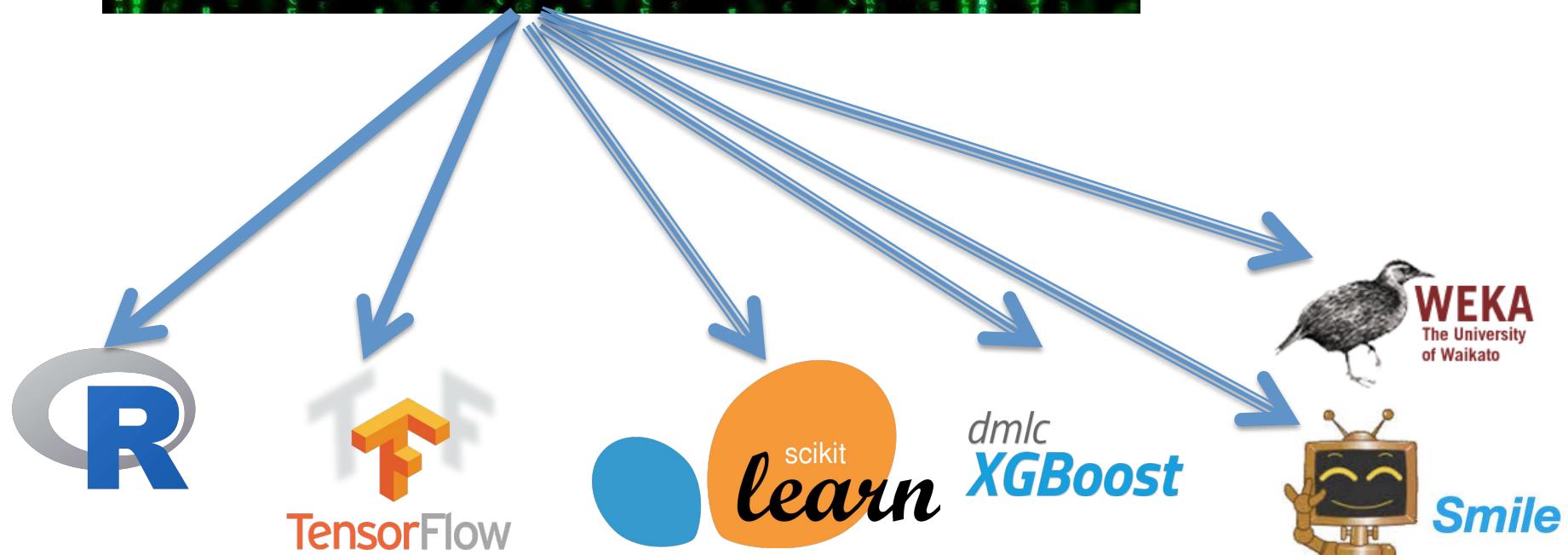
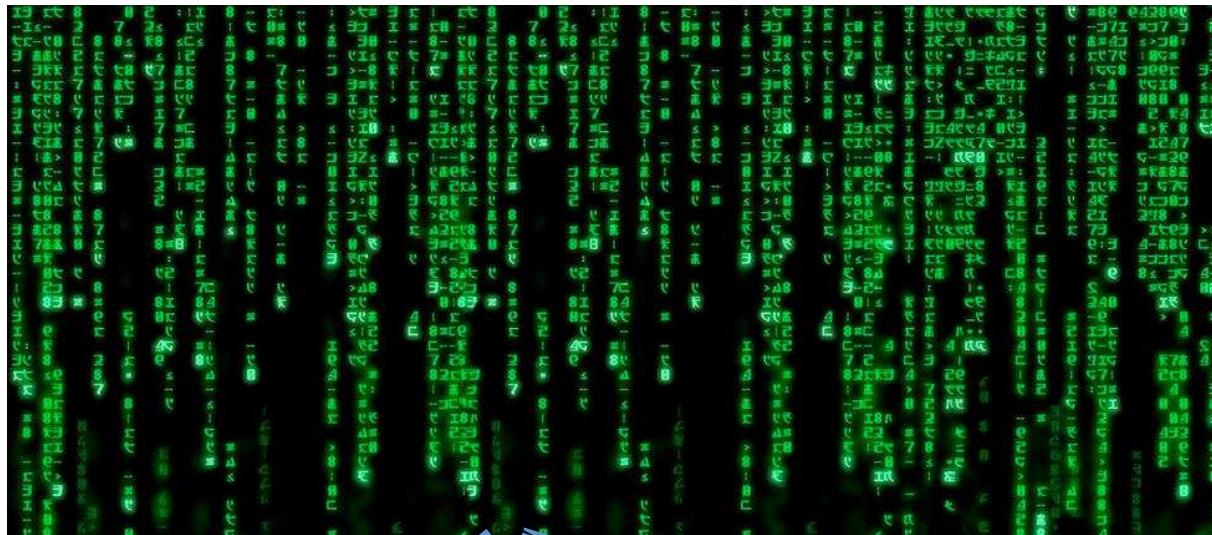
Product line and variability everywhere

- A video generator is a product line
 - Generalization of Bref generator
 - It is even a configurable generator
- Xtext is a configurable generator (see MWE2)
- JHipster is a product line
- ffmpeg is a product line
- MML language is a product line

JHipster

Case Study

MML Language



Variants and configurations

Other references

- Krzysztof Czarnecki and Ulrich Eisenecker "Generative Programming: Methods, Tools, and Applications"
- S. Apel, D. Batory, C. Kästner, and G. Saake. Feature-Oriented Software Product Lines: Concepts and Implementation. Berlin/Heidelberg: Springer-Verlag, 2013.
- Cory Kapser, Michael W. Godfrey: "Cloning considered harmful" considered harmful: patterns of cloning in software. Empirical Software Engineering 13(6): 645-692 (2008)
- C. Kästner. Virtual Separation of Concerns: Toward Preprocessors 2.0. PhD thesis, 2010
- Klaus Pohl, Günter Böckle, Frank van der Linden: Software Product Line Engineering - Foundations, Principles, and Techniques. Springer 2005

Other references

- Krzysztof Czarnecki, Krzysztof Pietroszek: Verifying feature-based model templates against well-formedness OCL constraints. GPCE 2006: 211-220
- José A. Galindo, Mauricio Alferez, Mathieu Acher, Benoit Baudry, and David Benavides. A Variability-based Testing Approach for Synthesizing Video Sequences (2014). In ISSTA'14
- Sarkar, A., J. Guo, N. Siegmund, S. Apel, and K. Czarnecki, "Cost-Efficient Sampling for Performance Prediction of Configurable Systems" In ASE'2015
- Mathieu Acher, Guillaume Bécan, Benoit Combemale, Benoit Baudry, and Jean-Marc Jézéquel. Product lines can jeopardize their trade secrets (2015). In ESEC/FSE'15