

Model-based Software Product Lines Overview and Principles

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

Material

<https://github.com/acherm/teaching-MDE-MIAGE1819>

<http://teaching.variability.io>

<http://familiar.variability.io>



Generator
~ composition of
video sequences



**video
variants**



Generator
~ composition of
video sequences

**video
variants**





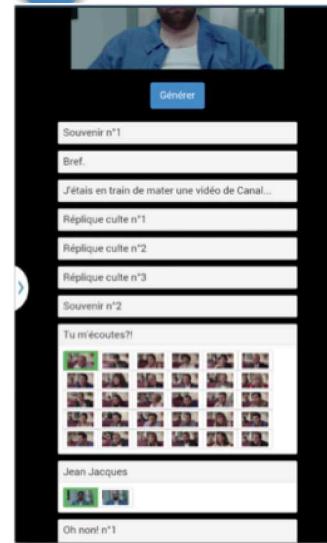
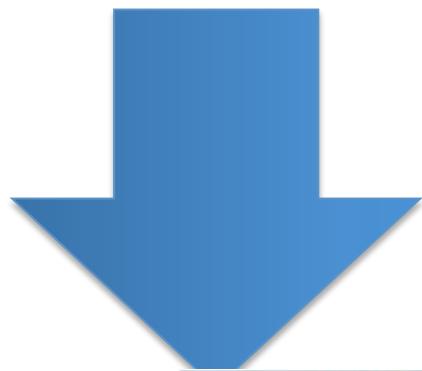
```

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"
```

Feature model: another model for modeling “features” of your Web site (eg ability to save the video; mode=generation with frequencies)

configurable
generator
of video generator

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

- Rappel Ingénierie des Exigences
 - Septembre 2017
 - Software Product Line Engineering
 - Feature Models
 - Defacto standard for modeling product lines and variability
- VaryLaTeX case
- 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

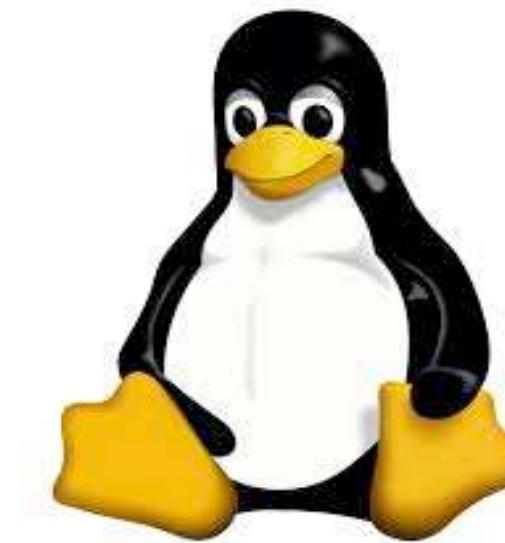
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <V> includes, <X> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module capable <> module capable

[] 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





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

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

THE PUPPY LICKS THE CAT
THE CAT LICKS THE PUPPY

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	✓	✓	✓	✓

[RENAULT](#) | [RENAULT VANS](#) | [CAR](#) | [VAN](#) | [ELECTRIC VEHICLES](#) | [RENAULT BUSINESS](#) | [USED CARS](#) | [OWNER SERVICES](#) | [ABOUT RENAULT](#) | [RENAULT SHOP](#) | [NEWS](#)

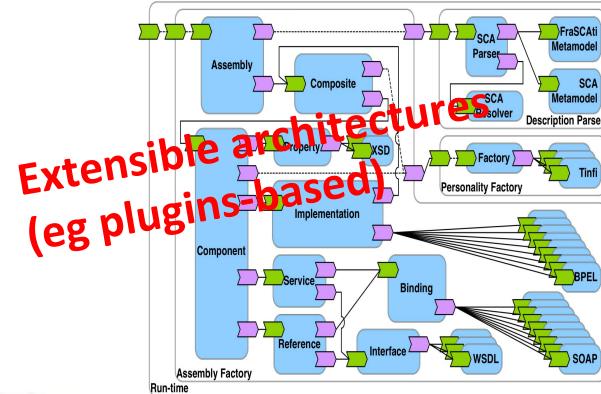
Renault UK | Renault Vans | New Kangoo Van Range | Kangoo Van | Build your own Kangoo Van | Selected Options

The screenshot shows the Renault Kangoo Van Range configurator. The top navigation bar includes 'NEW KANGOO VAN RANGE', '1 Preferences', 'Eqpt & Options', 'Previous' (disabled), and 'Next'. A large red watermark 'Configurators' is overlaid across the page. The main content area displays the following options:

<input checked="" type="checkbox"/>	COMFORT	
Central storage console & armrest between seats		£50.00
<input checked="" type="checkbox"/>	DRIVING	
Electric door mirror		£3.00
<input checked="" type="checkbox"/>	SAFETY & SECURITY	
ECS (Electronic Stability Control) with traction and understeer control		£200.00

To the right of the table is a small image of a white Renault Kangoo van.

The screenshot shows the 'System Preferences' window with the 'General' tab selected. On the left is a sidebar with a tree view of preference categories. The 'General' category is expanded, showing options like Appearance, Editors, Keys, Perspectives, Search, Security, Startup and Shutdown, Web Browser, and Workspace. Other collapsed categories include Art, Help, Java, LPG, Install/Update, Java, LPG, Plug-in Development, Run/Debug, Test, Team, Usage Data Collector, Validation, and XML. The main pane displays settings for the General tab, including checkboxes for 'Always run in background', 'Keep next/previous editor, view and perspectives dialog open', and 'Show heap status'. Below these is a section for 'Open mode' with radio buttons for 'Double click', 'Single click', 'Select on hover', and 'Open when using arrow keys'. A note at the bottom states: 'Note: This preference may not take effect on all views'. At the bottom right are buttons for 'Restore Defaults', 'OK', and 'Cancel'.



Extensible architecture
(eg plugins-based)



External Variability

Internal Variability

Being taught

Initial → Teach → Define → Work → School Break

Structural or behavioral models

```
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 Prod

DefaultType text/plain
AddDefaultCharset ISO-8859-1

UseThreadLocalStorage 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>
```

```
Configuration  
files
```

The screenshot shows a terminal window titled 'karmena - Pedowina' running a Linux kernel configuration tool. The command 'make menuconfig' has been entered. The configuration screen displays the 'Processor type and features' section of the kernel build system. The menu includes options like 'Processor family (Generic x86-64)', 'Preemption Model (No Preemption (FIFO))', 'Performance counter support', 'Machine check error reporting', 'Bell lapcounter support', 'KASLR (Kernel Address Space Layout Randomization)', 'Microcode support', 'CPU hotplug support', 'CPU information support', 'CPU thermal management support', 'Sparse Memory virtual memory (NLS)', 'Intel(R) Celeron(R) processor support', 'Intel(R) Core(TM) processor support', 'Intel(R) Xeon(R) processor support', 'Intel(R) Atom(TM) processor support', 'Reserve low 64G of RAM on ARM/Phoenix BIOS', 'ARM (Memory Type Range Register) support', 'PTE Clean Type support', 'PTE Clean support', 'Enable seccomp to safely compute untrusted bytecode', 'Enable fstack-protector buffer overflow detection (BNDT) support', 'Processor family (PPC (32/64 bit))', and 'Hexedit system call'. At the bottom of the screen, there are navigation keys: <select>, <exit>, <help>, and <back>. A large watermark reading 'Build system' is overlaid across the bottom right of the configuration interface.

list support
support

user allocation
#ix BIOSEN
support

trusted bytecode

Build systems

The screenshot shows the Eclipse IDE interface with two open files: `Notepad.java` and `Main.java`. The `Notepad.java` file contains Java code for a `Notepad` class, which includes methods for setting text and updating a `QuarkPanel`. The `Main.java` file has a single line of code: `new Notepad().setVisible(true);`. On the right side of the interface, there is an `AST View` window displaying the Abstract Syntax Tree (AST) for the selected code. The tree structure shows nodes for `MethodDeclaration`, `BlockStatement`, `ForStatement`, `MethodInvocation`, and `AssignmentStatement`, among others. The `AST View` also includes a toolbar with various icons for navigating the tree.

```
Notepad.java
Main.java 23

public void setText(String t) {
    program.setText(t);
    equation.setText("");
    updatQuarkPanel();
}

apply.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (hoa.isSelected()) {
            t = t.appendln(hoa.getText() + "\n");
        }
        if (gadvice.isSelected()) {
            t = t.appendln(gadvice.getText() + "\n");
        }
        if (intro.isSelected()) {
            t = t.appendln(intro.getText() + "\n");
        }
        if (gadvice.isSelected()) {
            t = t.appendln(gadvice.getText() + "\n");
        }
        if (hoa.isSelected() || gadvice.isSelected() || intro.isSelected()) {
            hoa.setSelected(false);
            gadvice.setSelected(false);
            intro.setSelected(false);
        }
    }
})
```



(a) Variant #1 of video sequence



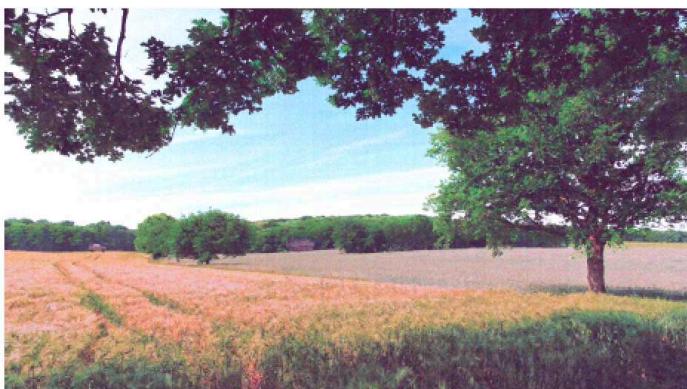
(b) Variant #2 of video sequence



(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

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



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

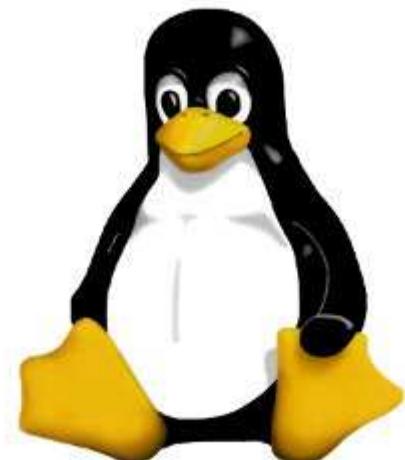
320^{optional, independent}
features

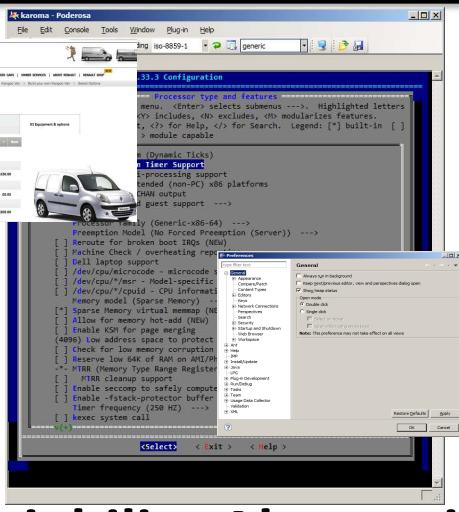
more variants than estimated
atoms in the universe



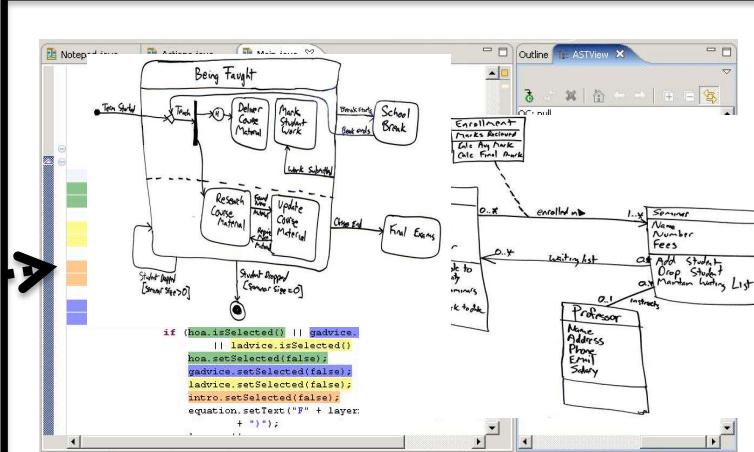
2000 features

10000
features





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



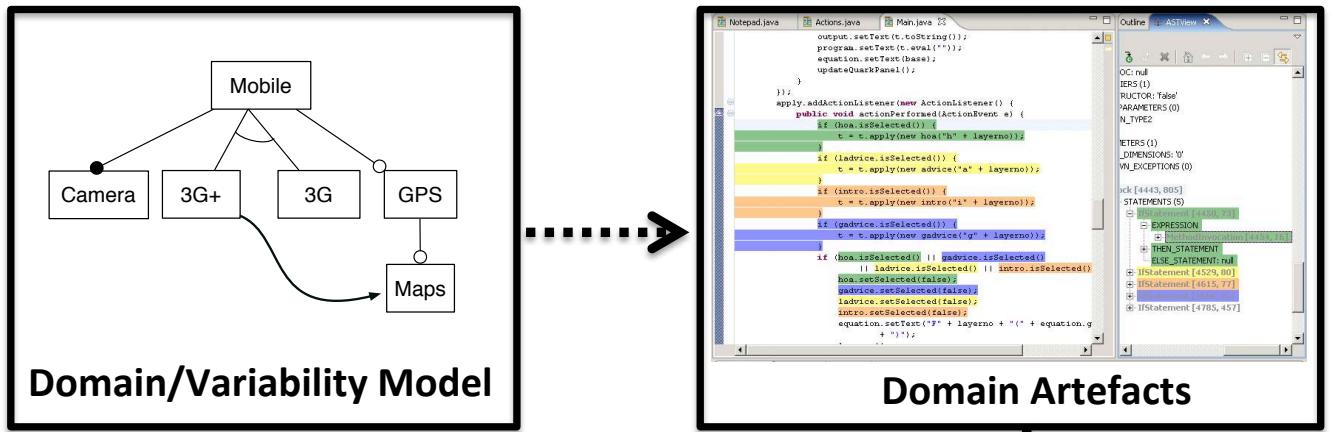
mapping

```
1 package io.jhipster.generator.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.hazelcast.HazelcastCacheManagerFactory;
10 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.Autowired;
11 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
12 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
13 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
14 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
15 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
16 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
17 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
18 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
19 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
20 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
21 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
22 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
23 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
24 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
25 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
26 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
27 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
28 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
29 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
30 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
31 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
32 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
33 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
34 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
35 import org.hibernate.cache.hazelcast.HazelcastCacheManagerFactory.AutoWired;
```

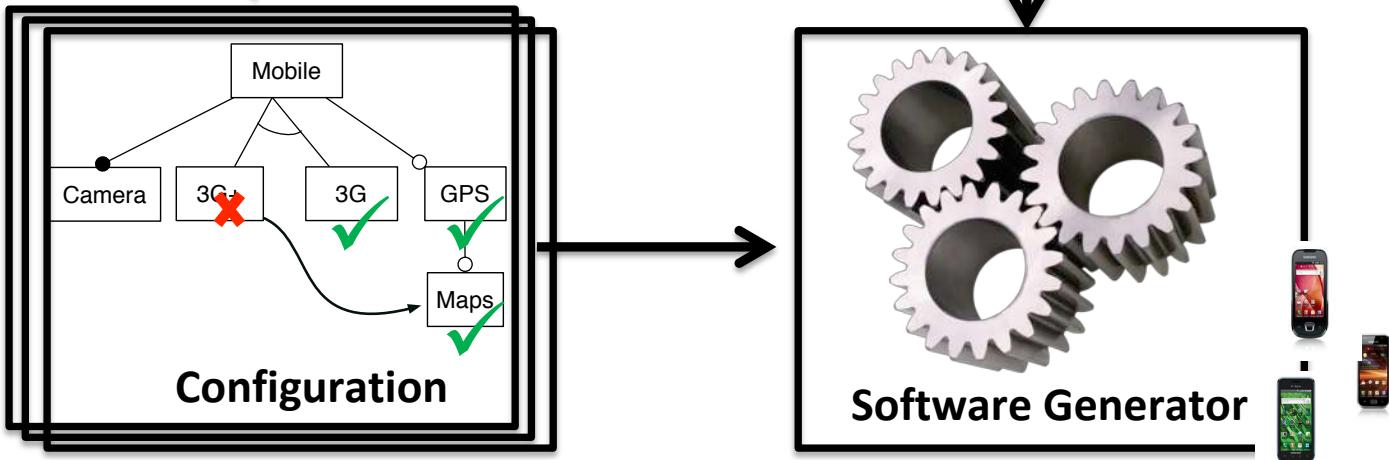
Base Artefacts

Software Generator (derivation engine)

Domain Engineering



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** »

Jan Bosch et al. (2004)

« 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
```

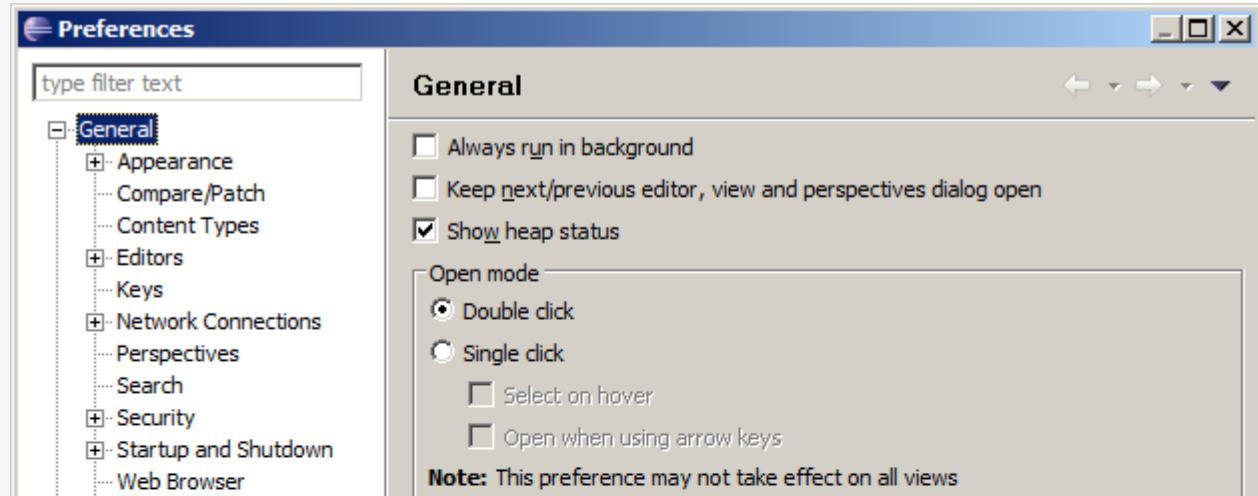
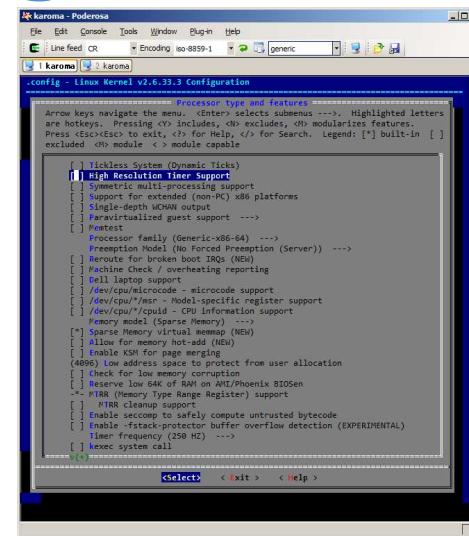
```
HostnameLookups Off
```

```
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

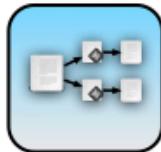
Code Cloning (example, Linux driver)

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;  
}  
.....
```

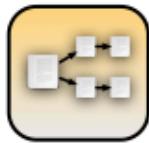


Replicate & Specialize

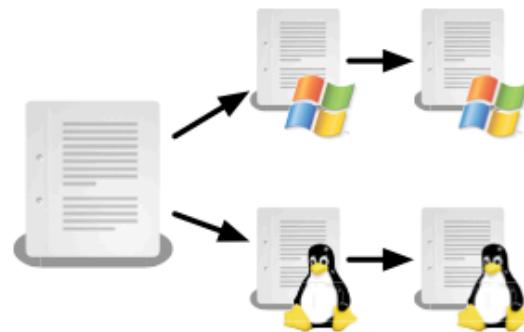


Clone to reuse and adapt existing solutions

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

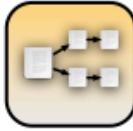


Platform Variations

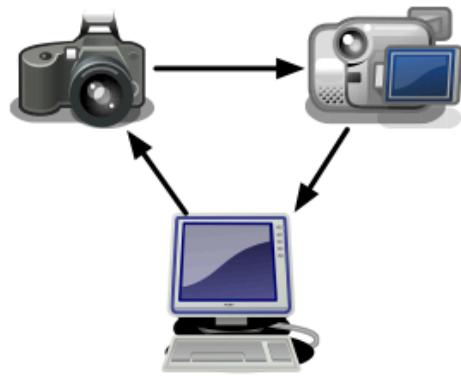


**Clone existing code and fix
low level platform interaction**

- + 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

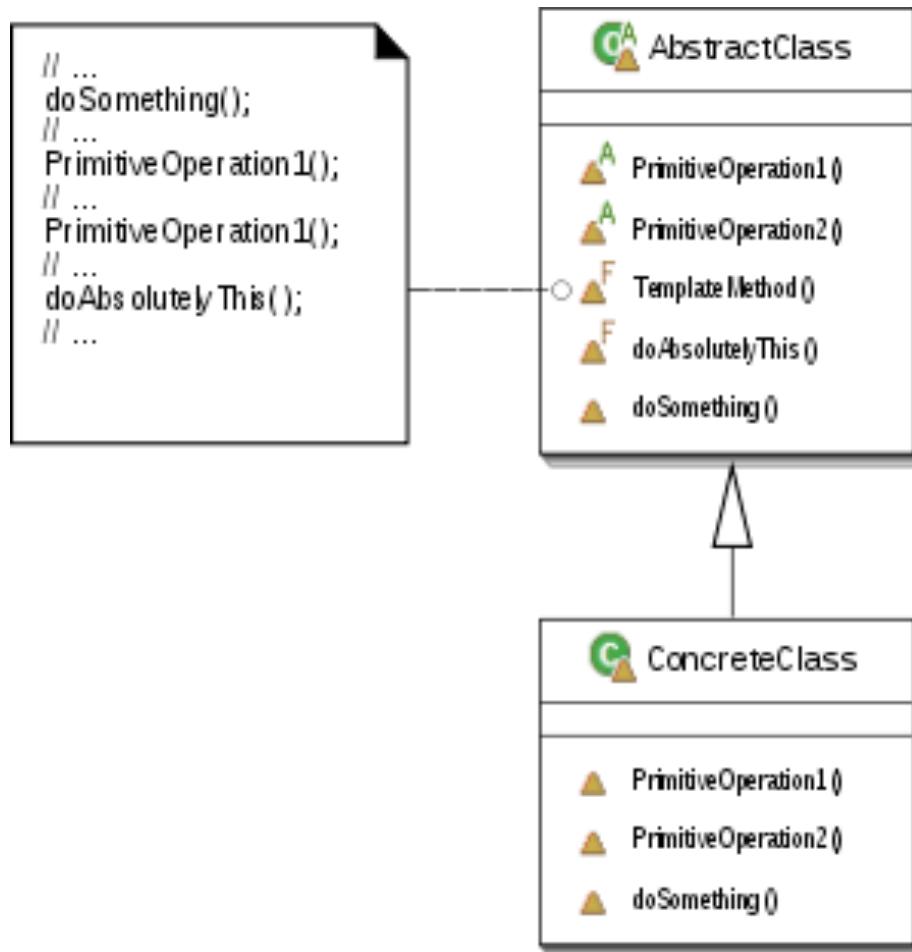
Factory

Strategy

Decorator

....

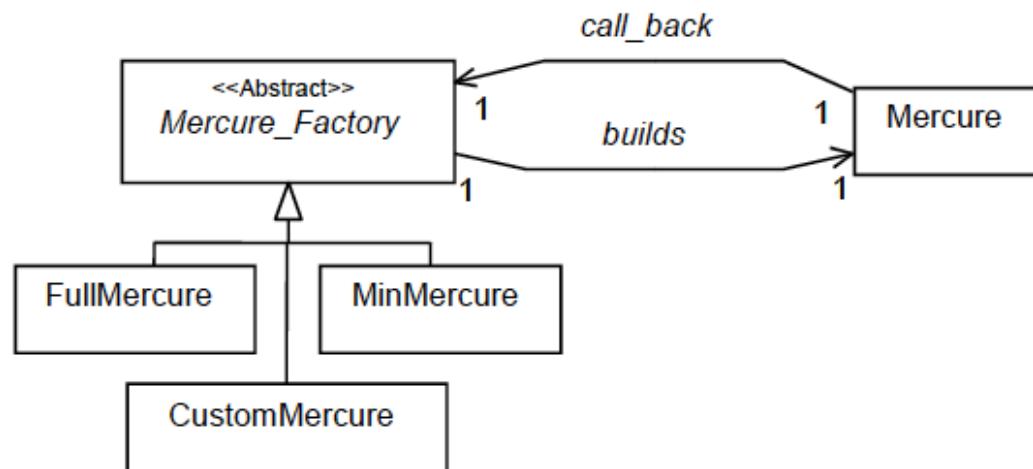
Template Method



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

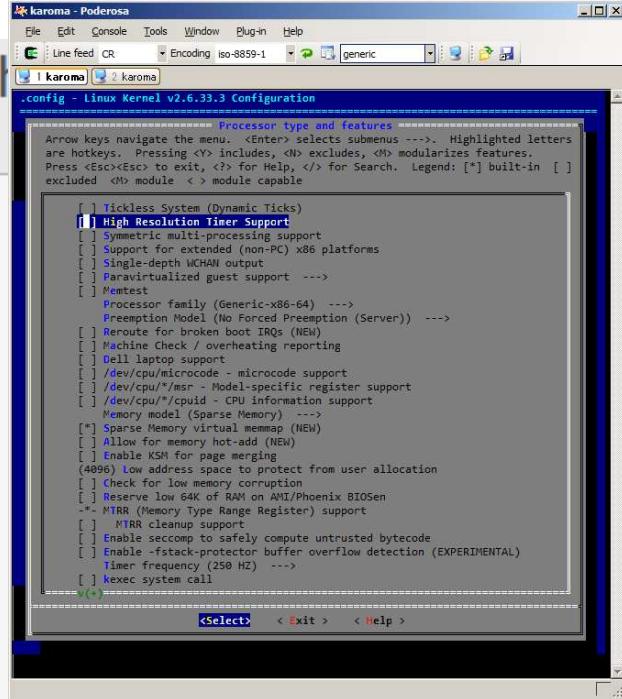
ErrorLog logs/error.log
LogLevel error

PidFile logs/httpd.pid

Timeout 300

KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

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



A photograph of a heavily rusted, vintage-style pickup truck. The truck is positioned diagonally, facing towards the top left. It has a flatbed bed and appears to be in a state of disrepair, with its bodywork and paint long gone. The background consists of a dense, overgrown hillside covered in green and brown vegetation.

Unused flexibility



Illegal variant

Feature Models

A screenshot of the karmo configuration tool. It displays a list of kernel features, each with a checkbox. Some features have descriptive text next to them. The features include: Processor type and features (Processor family (Generic<-x86-64) ...), Preemption Model (No Forced Preemption (Server) ...), Memory model (Sparse Memory) ..., Allow for memory hot-add (NDA) ..., Enable KSM for page merging, Check for low memory corruption, Reserve low 64K of RAM on AMI/Phoenix BIOSen, and others. At the bottom, there are buttons for <Select>, <Exit>, and <Help>.

Variability Model

A screenshot of an IDE showing Java code. The code is annotated with various colors (green, yellow, orange, blue) and symbols, likely representing the variability model. The code is part of a class named Notepad.java and includes imports for Actions.java and Main.java. It contains several if statements and method calls like apply.addActionListener and t.apply. The annotations highlight specific parts of the code, such as variable names and method parameters, which correspond to the variability model defined in the first window.

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

is crucial



Configuration



Software Generator

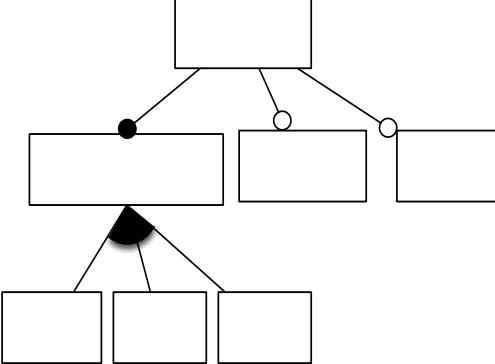
Unused flexibility





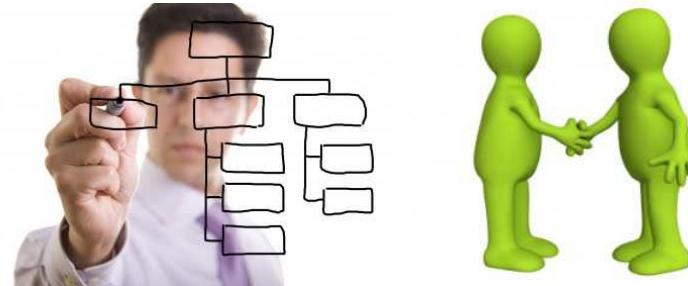
Illegal variant

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

Habitacle

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

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

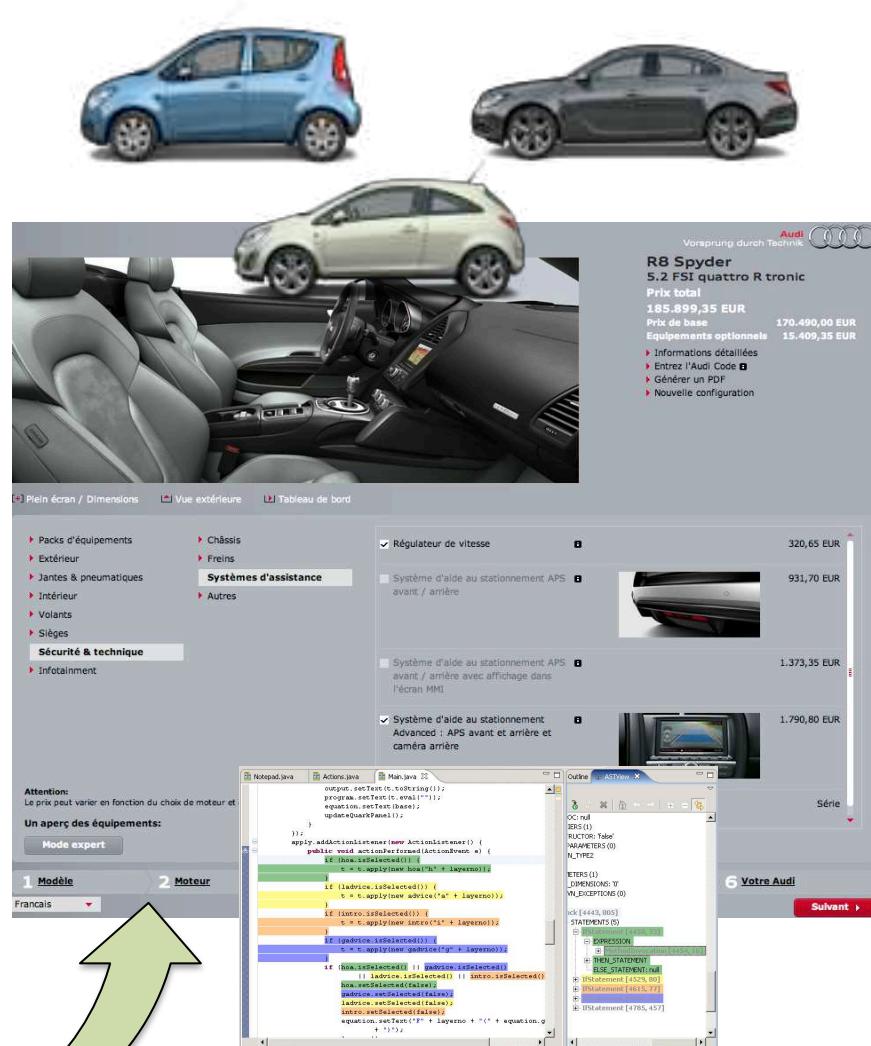
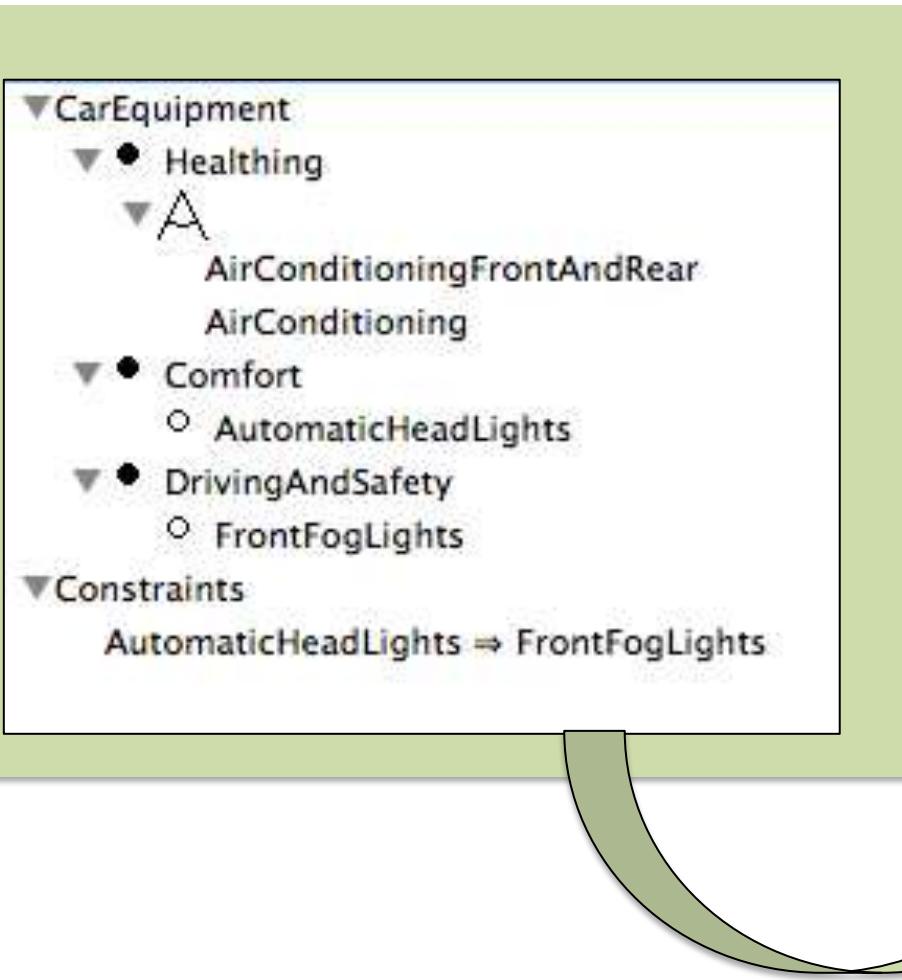
5 Option

6 Votre Audi

Français

Suivant ▶

Feature Models



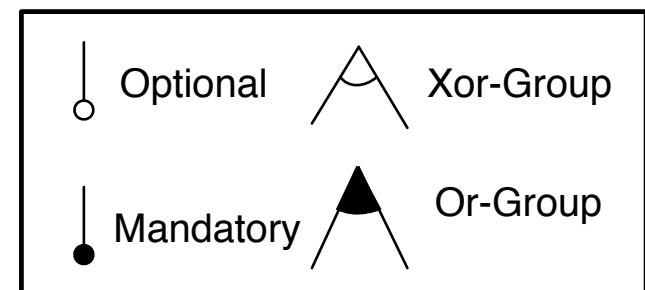
Feature Models (Background)

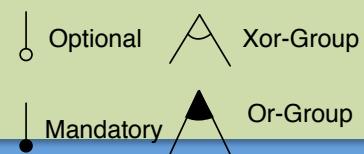
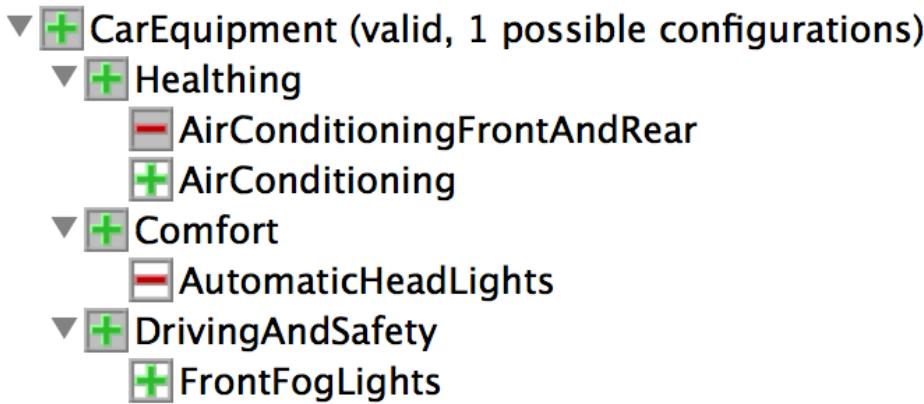


Hierarchy: rooted tree

Variability:

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





Hierarchy + Variability

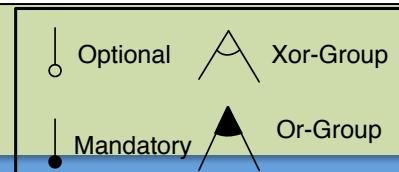
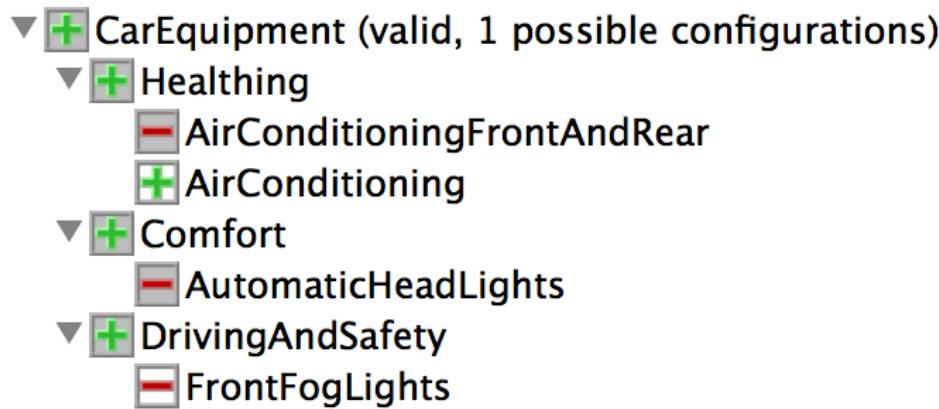
=

set of valid configurations

configuration = set of features selected

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





Hierarchy + Variability

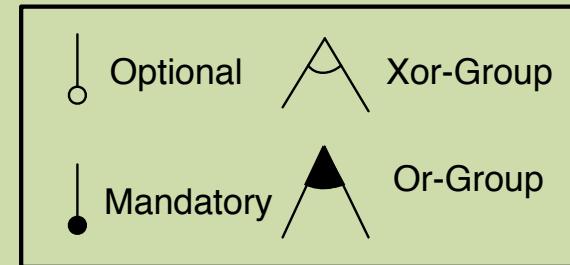
=

set of valid configurations

configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning}





Hierarchy + Variability

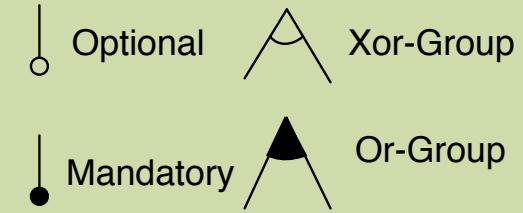
=

set of valid configurations

configuration = set of features selected

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



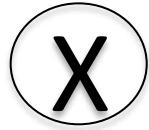


Hierarchy + Variability

=

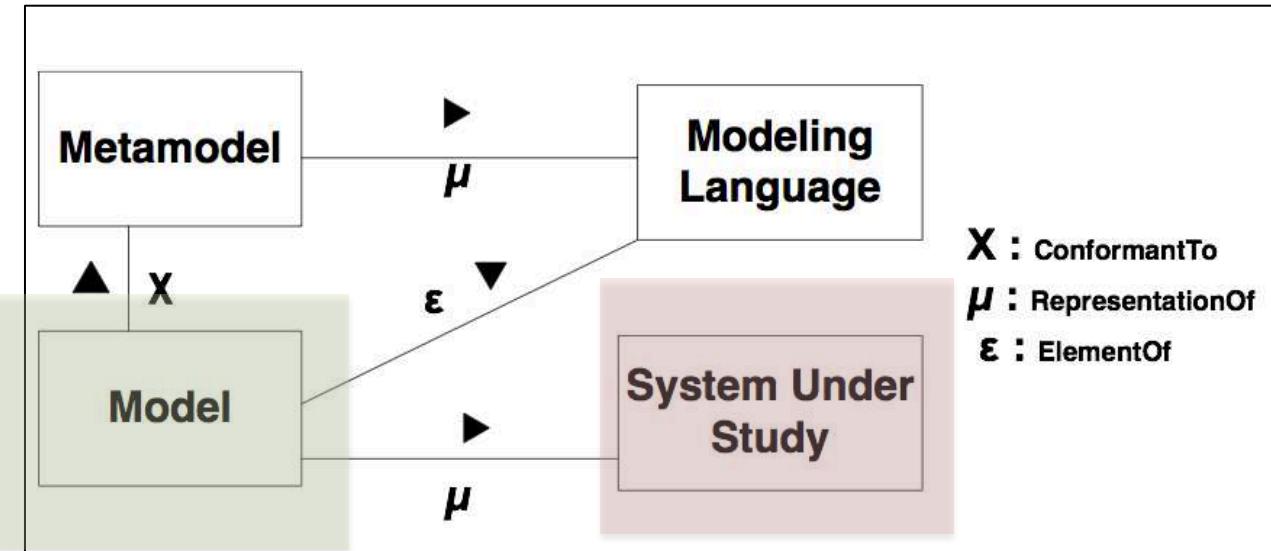
set of valid configurations

{CarEquipment, Comfort,
DrivingAndSafety,
Healthing}



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

Feature Models

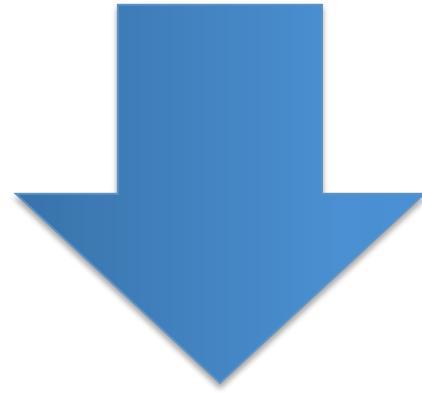




```
foo1.videogen ✘

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"
}

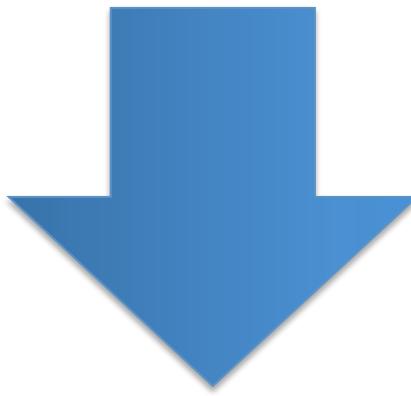
alternatives v4 {
    videoseq v41 "v4/seq1.mp4"
    videoseq v42 "v4/seq1.mp4"
}
mandatory videoseq v5 "https://www.youtube.com/watch?v=ezKx-S0LiNQ"
```



 FFmpeg

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

```
VideoGen {  
    mandatory videoseq v1 "V1/v1.mp4"  
    optional videoseq v2 "v2folder/v2.mp4"  
    alternatives v3 {  
        videoseq v31 "v31.mp4"  
        videoseq v32 "v32.mp4"  
    }  
}
```



model-to-text

```
fmVideoGen = FM (VideoGen: v1  
[v2] v3; v3: (v31|v32); )
```

FAMILIAR

Product line and variability everywhere

- Your video generator is a product line
 - Generalization of Bref generator
 - It is even a configurable generator
 - There is an hidden feature model in every VideoGen specification
- Xtext is a configurable generator (see MWE2)
- JHipster is a product line
- ffmpeg is a product line

JHipster

Case Study

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