

# Variability and Software Product Lines (an overview)

Mathieu Acher

Maître de Conférences

[mathieu.acher@irisa.fr](mailto:mathieu.acher@irisa.fr)

# Material

[https://github.com/FAMILIAR-project/  
HackOurLanguages-SIF](https://github.com/FAMILIAR-project/HackOurLanguages-SIF)

# Plan

- Challenges and Overview
  - Developping billions of software product is hard but now a common practice
- Implementing Variability
  - Revisit of existing techniques and curriculum
- Specificity of Product Line Engineering
  - Process, methods
- Feature Models
  - Defacto standard for modeling product lines and variability
  - Syntax, semantics, automated reasoning, synthesis

# Contract

- The idea of software product lines and variability
  - You will be able to recognize this class of systems
  - Aware of the complexity, the specific development process, and existing techniques
- Feature modeling
  - A widely used language for modeling product lines and configurable systems in a broad sense
- Composing/Decomposing feature models with a domain-specific language
- Reverse engineering variability models

A wide-angle photograph of a vast parking lot filled with thousands of vehicles, primarily white sedans and compact cars, arranged in long, horizontal rows. Interspersed among them are several white and light-colored pickup trucks. The parking lot is bounded by a fence in the background, and a body of water is visible beyond the fence. The sky is clear and blue.

# Product Lines

# (Software) Product Lines



01011011  
110111110  
001101110  
1100011101  
1000011111  
101001110  
100001010  
101001011  
000001110  
11010101  
111011010  
011000100  
01010101  
110101010  
1010101010

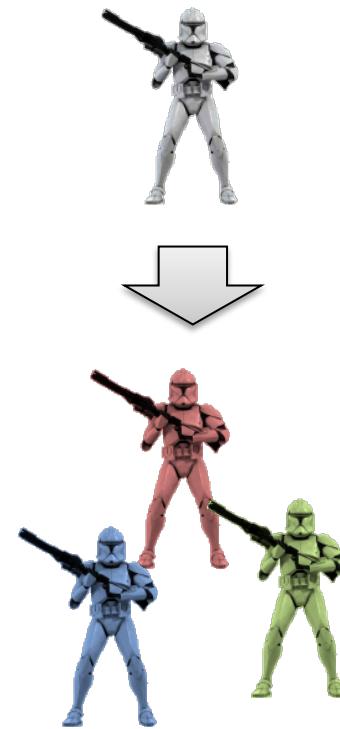


# The three ways to build a (software) product

Independently



„Clone & Own“



„Shared“ (reusable) Assets



(credits: Thorsten Berger's slide)

# The three ways to build a (software) product

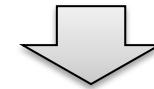
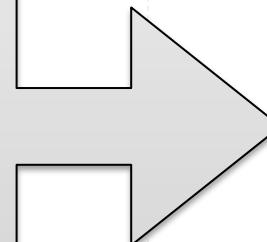
Independently

„Clone & Own“

„Shared“ (reusable) Assets

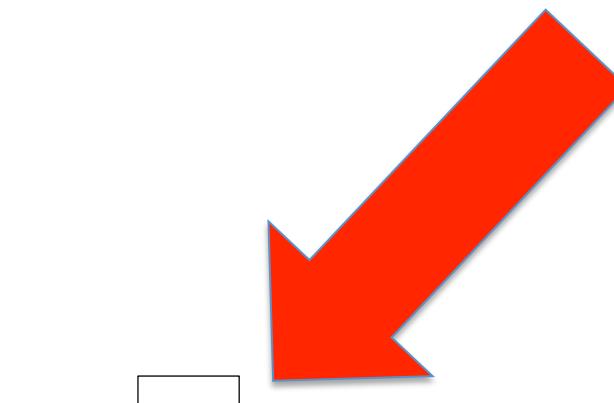
## Software Product Lines

Product Configuration  
Variability Modeling  
Components  
Domain-specific Languages  
Generators  
Preprocessors  
Design Patterns  
...

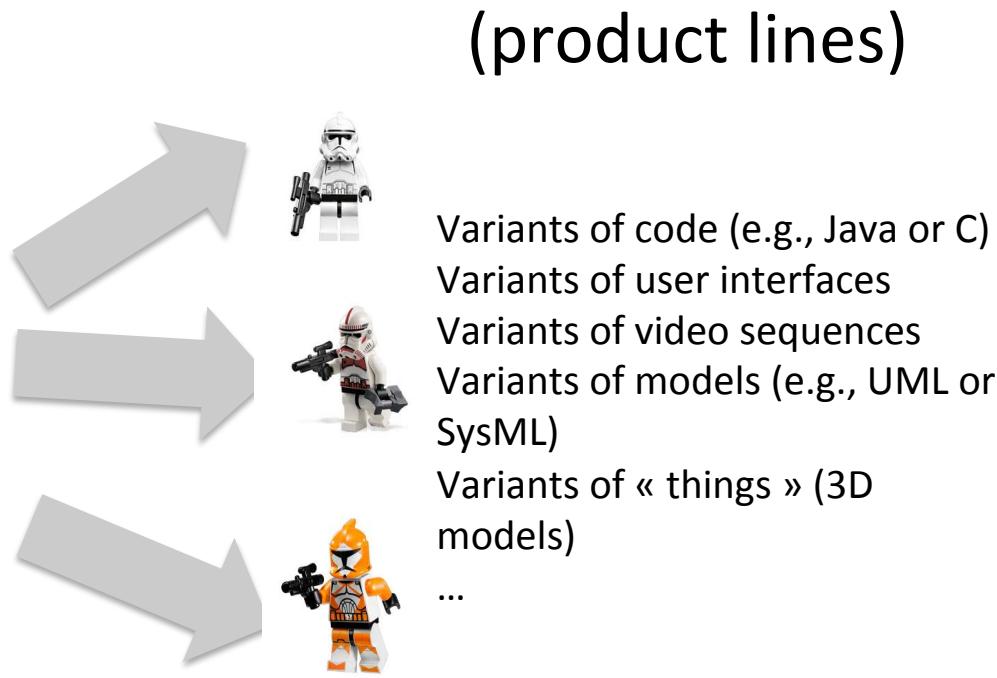


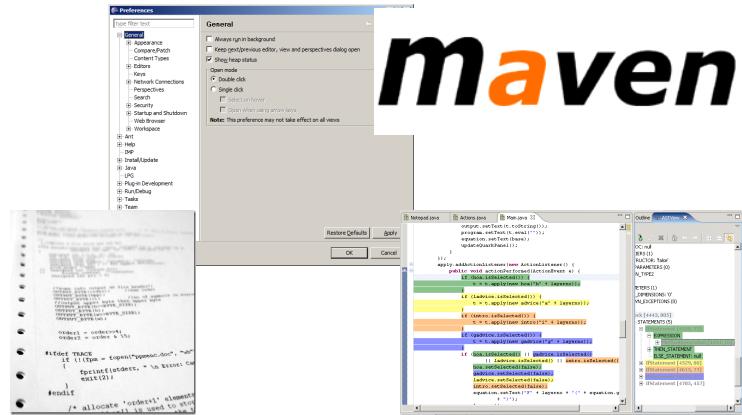
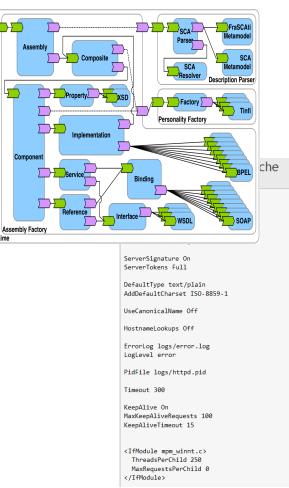
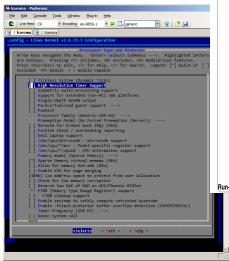
(credits: Thorsten Berger's slide)

# Modeling and Reverse Engineering Variability



Feature models  
or Product Matrices

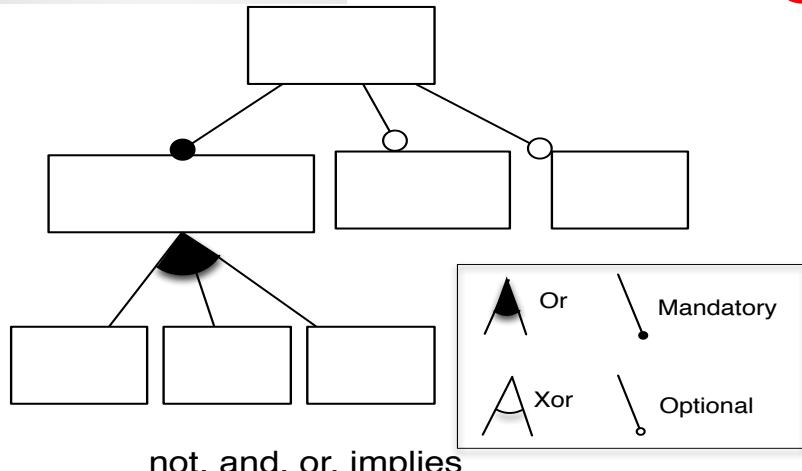




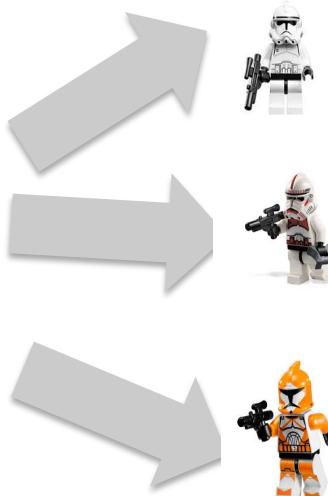
**maven**

# Mining/Extracting Encoding/Formalizing Synthesising

FAMiliAR

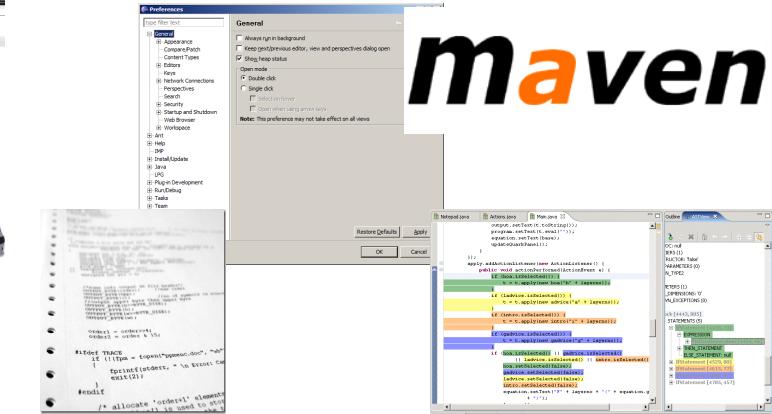
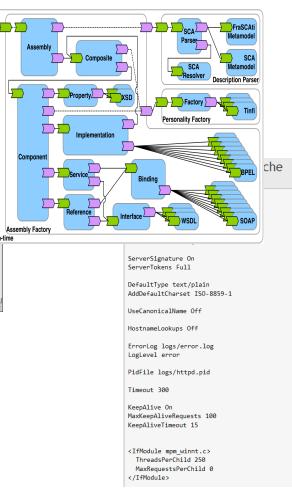
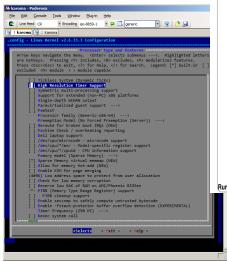


Variability Models (feature models)



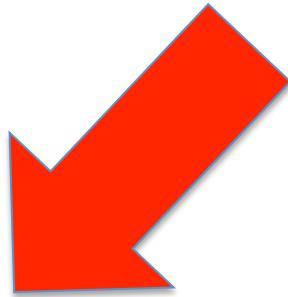
Variants of code (e.g., Java or C)  
 Variants of user interfaces  
 Variants of video sequences  
 Variants of models (e.g., UML or SysML)  
 Variants of « things » (3D models)

...



# maven

# Mining/Extracting Encoding/Formalizing Synthesising

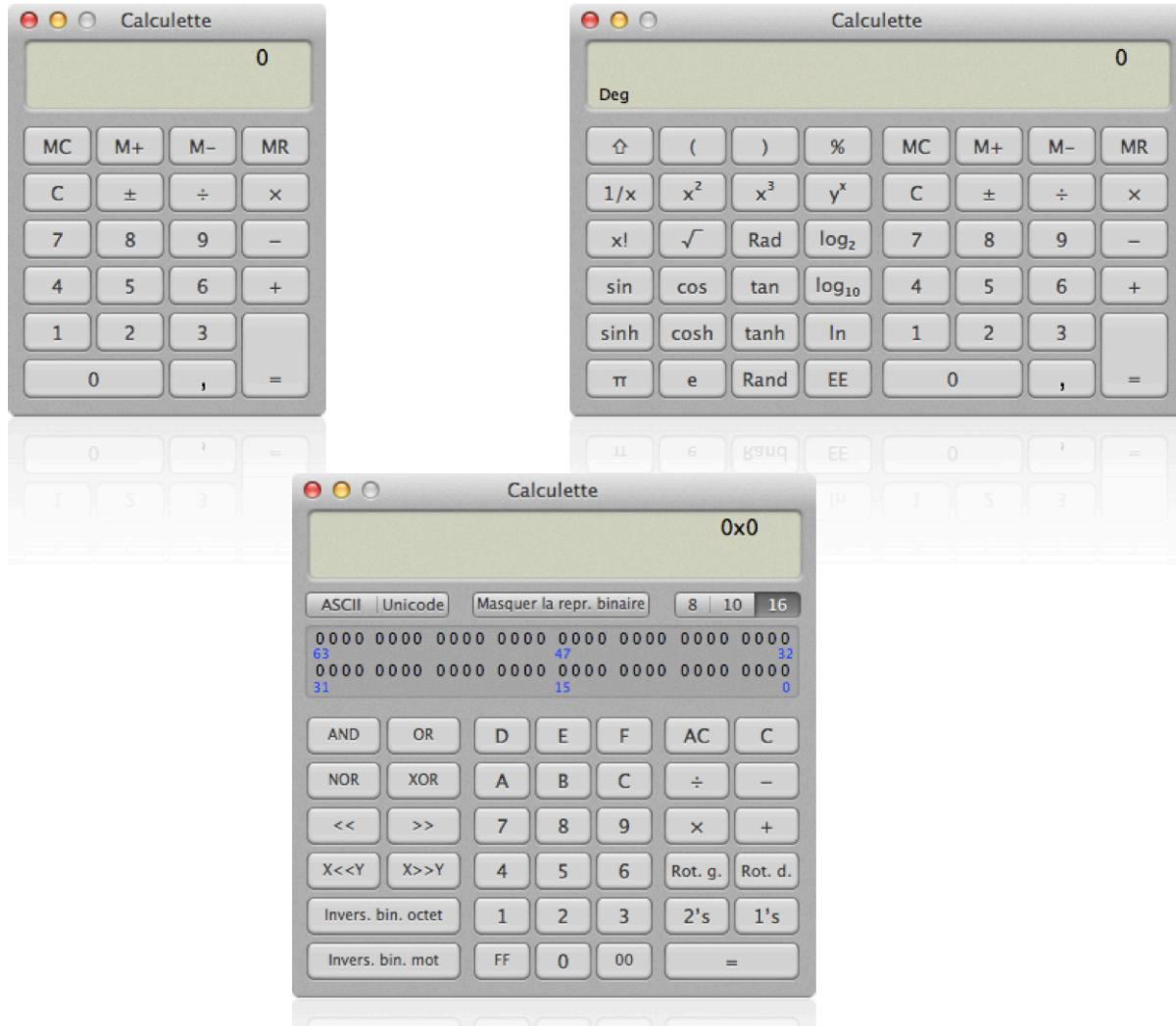


OpenCompare

Product	License	Price	Language Support	Language	WYSIWIG
W1	Commercial	10	Yes	Java	Yes
W2	NoLimit	20	No		Yes
W3	NoLimit	10	No		Yes
W4	GPL	0	Yes	Python	Yes
W5	GPL	0	Yes	Perl	Yes
W6	GPL	10	Yes	Perl	Yes
W7	GPL	0	Yes	PHP	No
W8	GPL	10	Yes	PHP	Yes

# Variability and Software Product Lines

Perhaps, you ignore the names of something omnipresent in numerous contexts



« A set of programs is considered to constitute a **family**, whenever it is worthwhile to study programs from the set by **first studying the common properties** of the set and then determining the **special properties** of the individual family members »



aka Variability

David L. Parnas — “On the design and development of program families” in Transactions on Software Engineering, SE-2(1):1–9, 1976

# Variability

“the ability of a system to be efficiently extended, changed, customized or configured for use in a particular context”

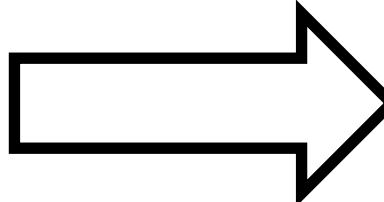
*Mikael Svahnberg, Jilles van Gurp, and Jan Bosch (2005)*







# Software-intensive systems



come in many variants



# RENAULT VANS



CARS | VANS | ELECTRIC VEHICLES | RENAULT BUSINESS | USED CARS | OWNER SERVICES | ABOUT RENAULT | RENAULT SHOP **NEW**

Renault UK > Renault Vans > New Kangoo Van Range > Kangoo Van > Build your own Kangoo Van > Select Options

## NEW KANGOO VAN RANGE

01 Preferences

02 Version

03 Equipment & options

< Previous

Next >

### OPTIONS

#### > COMFORT

- |   |        |
|---|--------|
| <input checked="" type="checkbox"/> Central storage console & armrest between seats | £50.00 |
|---|--------|

#### > DRIVING

- |  |       |
|--|-------|
| <input type="checkbox"/> Electric door mirrors | £0.00 |
|--|-------|

#### > SAFETY & SECURITY

- |   |         |
|---|---------|
| <input checked="" type="checkbox"/> ESC (Electronic Stability Control) with traction and understeer control | £200.00 |
|---|---------|



“Reverse Engineering Web Configurators” Ebrahim Khalil Abbasi, Mathieu Acher, Patrick Heymans, and Anthony Cleve. In 17th European Conference on Software Maintenance and Reengineering (CSMR'14)

# LE PLIAGE PERSONNALISÉ



MODÈLES

COULEUR  
RECTOCOULEUR  
VERSO

BOUCLERIE



RESET



LE PLIAGE CUIR

LE PLIAGE TOILE



## VOTRE PERSONNALISATION

Porte-monnaie Toile : 9 x 7 x 5 cm  
Couleur recto : Garance  
Couleur verso : Malabar  
Bouclerie : Bronze

35,00 € AJOUTER AU PANIER

Infos Partager J'aime

<a href="#">Developer Tools</a>
<a href="#">Development</a>
<a href="#">Drivers</a>
<a href="#">DTP/Prepress</a>
<a href="#">Educational</a>
<a href="#">Finance</a>
<a href="#">Font Tools</a>
<a href="#">Games</a>
<a href="#">Graphics</a>
<a href="#">HTML Tools</a>
<a href="#">Internet Utilities</a>
<a href="#">iPhone Applications</a>
<a href="#">iPod Tools</a>
<a href="#">Math/Scientific</a>
<a href="#">Multimedia</a>
<a href="#">Network/Admin</a>
<a href="#">Screensavers</a>
<a href="#">Security</a>
<a href="#">Spotlight Plugins &amp; Utilities</a>
<a href="#">System Utilities</a>
<a href="#">Utilities</a>
<a href="#">Video</a>
<a href="#">Word Processing</a>
 <a href="#">GLOBAL PAGES &gt;&gt;</a>
<a href="#">NEWS ARCHIVE &gt;&gt;</a>
<a href="#">DTPEDIA REVIEWS &gt;&gt;</a>
<a href="#">MEET THE EDITORS &gt;&gt;</a>

## Power Matte 2.0.1.3 update



Adobe After Effects plugin that can extract any object in an Image  
[\[read more >\]](#)

**Size:** 13.20 MB  
**Platform:** Mac OS X 10.5 or later  
**License:** Trial  
**Rating:** Good (3.0/5)  
**Downloads:** 1,504  
**Updated:** June 20th, 08:21 UTC

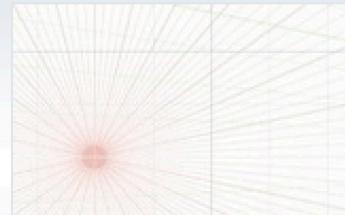


## Gridus 1.1 update



Helps you generate perspective grids  
[\[read more >\]](#)

**Size:** 102 KB  
**Platform:** Mac OS X 10.8 or later  
**License:** Commercialware  
**Rating:** NOT RATED  
**Downloads:** 21  
**Updated:** June 20th, 07:56 UTC



## Picture Frame 2.2 update



Quickly generate multi-frame photos using your Mac  
[\[read more >\]](#)

**Size:** 716 KB  
**Platform:** Mac OS X 10.6.6 or later  
**License:** Commercialware  
**Rating:** Excellent (5.0/5)  
**Downloads:** 297  
**Updated:** June 20th, 07:53 UTC

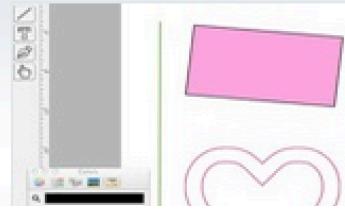


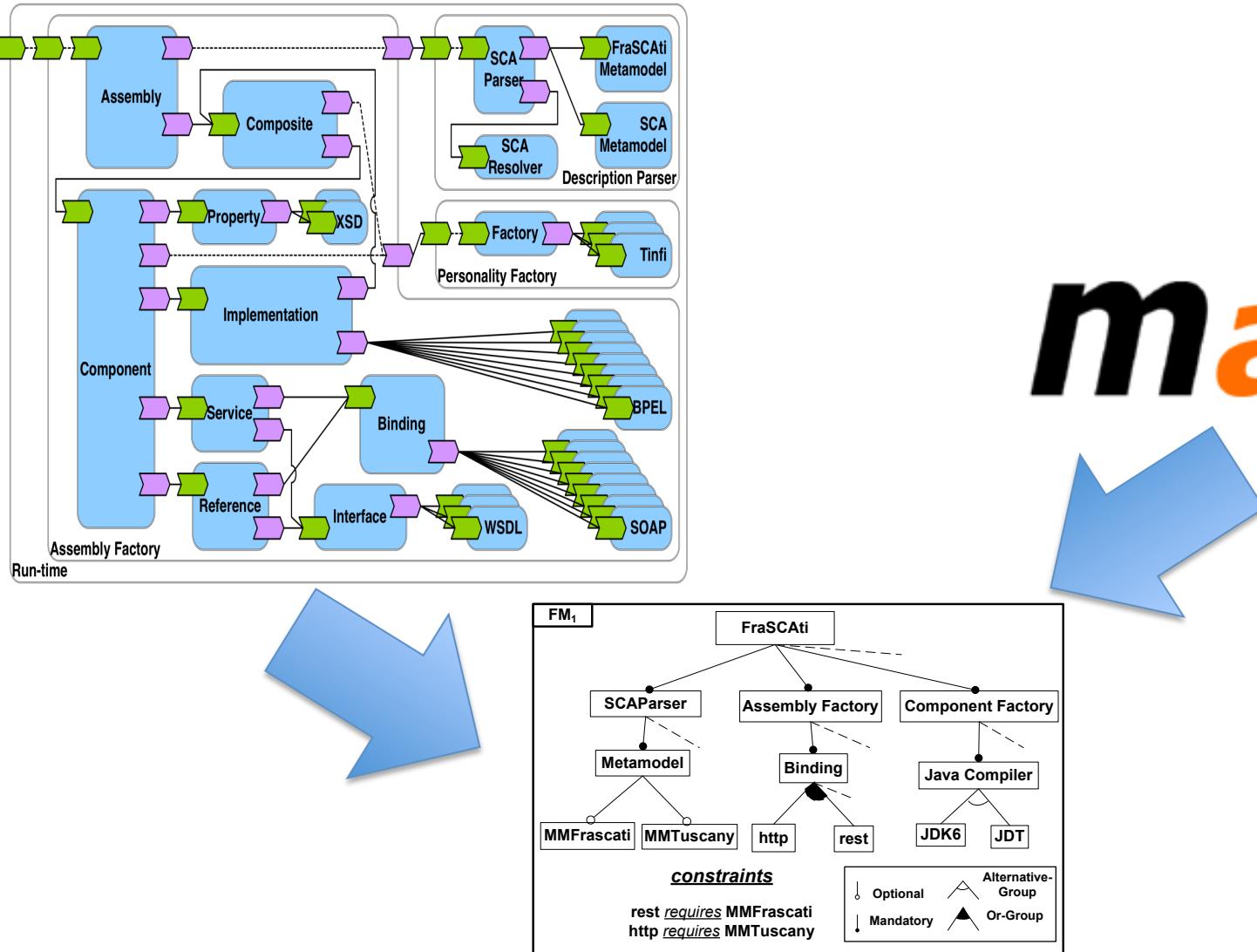
## FashionLab Studio 1.1 update



Makes it easy to design your own T-shirt using a Mac  
[\[read more >\]](#)

**Size:** 3.10 MB  
**Platform:** Mac OS X 10.6.6 or later  
**License:** Commercialware  
**Rating:** NOT RATED  
**Downloads:** 3  
**Updated:** June 20th, 07:49 UTC



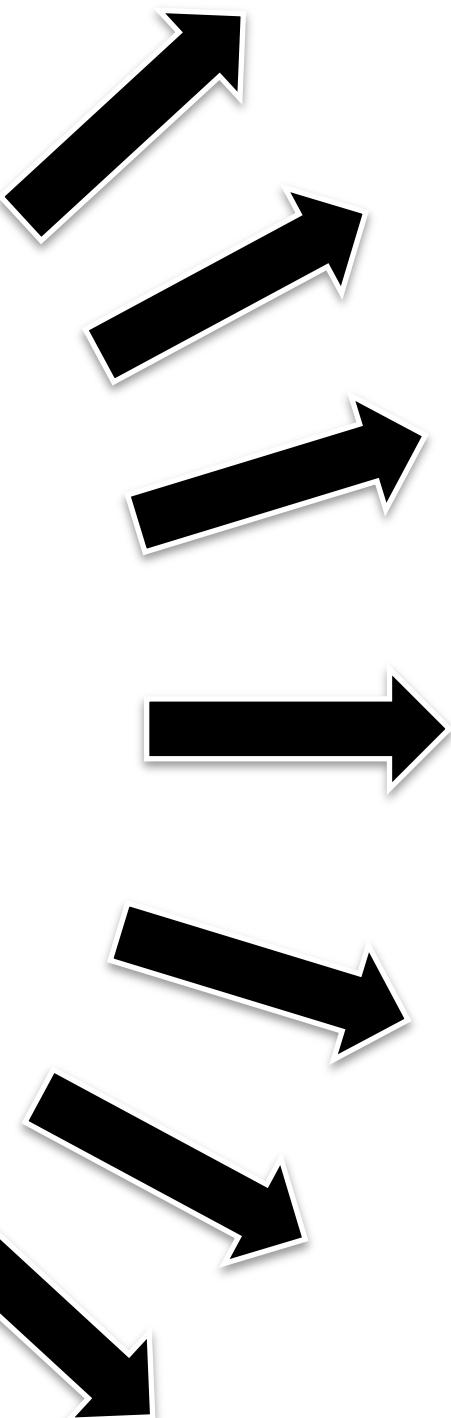


# maven

## Variability Model

« Extraction and Evolution of Architectural Variability Models in Plugin-based Systems »  
 Mathieu Acher, Anthony Cleve, Philippe Collet, Philippe Merle, Laurence Duchien, Philippe Lahire ECSA/SoSyM'14

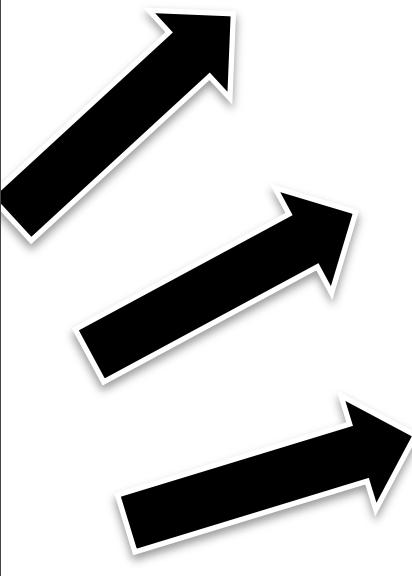
# Printer Firmware



```
1 karmo 2 karmo
Line Feed CR Encoding iso-8859-1 generic .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 <> includes, <> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded [<> module] < > module capable
[*] Tickless System (Dynamic Ticks)
[ ] High Resolution Timer Support
[ ] SYSENTER/SYSEXIT handling support
[ ] Support for extended (non-PC) x86 platforms
[ ] Single-depth ICHAN output
[ ] Paravirtualized guest support ...
[ ] Memtest
[ ] Processor family (Generic-x86-64) --->
[ ] Preemption Model (No Forced Preemption (Server)) --->
[ ] Renote 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 64M 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
v(<)

    <Select> < Exit > < Help >
```

# Linux Kernel



Brand	Model name	Sensor size	Effective megapixels	Lens mount	Viewfinder type	Viewfinder coverage (% of the frame)	Metering zones	Focus points	Lowest ISO	Highest ISO	DxOMark sensor score	DxO ISO performance <sup>[1]</sup>	
Canon	1D X	Full frame	18.1	EF	Pentaprism	100	252	61	50	204800	82	2786	
Canon	1Ds Mark III	Full frame	21.1				63	45	50	3200	80	1663	
Canon	1D Mark IV	APS-H	16.1				63	45	50	102400	74	1320	
Canon	5D Mark III	Full frame	22.3				63	61	50	102400	81	2293	
Canon	5D Mark II	Full frame	21.1				35	9	50	25600	79	1815	
Canon	6D	Full frame	20.2				63	11	100	102400	82	2340	
Canon	7D	APS-C	18.0				63	19	100	12800	66	854	
Canon	70D	APS-C	20.2				63	19	100	25600	68	926	
Canon	60D	APS-C	18.0				63	9	100	12800	66	813	
Canon	50D	APS-C	15.1		EF, EF-S	Pentaprism	95	35	9	100	12800	63	696
Canon	40D	APS-C	10.1		EF, EF-S	Pentaprism	95	35	9	100	3200	64	703
Canon	30D	APS-C	8.2		EF, EF-S	Pentaprism	95	35	9	100	3200	59	736
Canon	20D	APS-C	8.2		EF, EF-S	Pentaprism	95	35	9	100	3200	62	721



Guillaume Bécan, Nicolas Sannier, Mathieu Acher, Olivier Barais, Arnaud Blouin, and Benoit Baudry.  
 Automating the Formalization of Product Comparison Matrices (2014). In 29th IEEE/ACM International Conference on Automated Software Engineering (ASE'14)



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

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

```
JHIPSTER GENERATOR
  JHipster
    JHipster
  JHipster
  JHipster
```

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)
```

[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') { %>
```

# Variability

```

class Graph {
    Vector nv = new Vector(); Vector ev = new Vector();
    Edge add(Node n, Node m) {
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
        e.weight = new Weight();
        return e;
    }
    Edge add(Node n, Node m, Weight w) {
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
        e.weight = w; return e;
    }
    void print() {
        for(int i = 0; i < ev.size(); i++) {
            ((Edge)ev.get(i)).print();
        }
    }
}

```

```

class Node {
    int id = 0;
    Color color = new Color();
    void print() {
        Color.setDisplayColor(color);
        System.out.print(id);
    }
}

```

```

class Edge {
    Node a, b;
    Color color = new Color();
    Weight weight = new Weight();
    Edge(Node _a, Node _b) { a = _a; b = _b; }
    void print() {
        Color.setDisplayColor(color);
        a.print(); b.print();
        weight.print();
    }
}

```

```

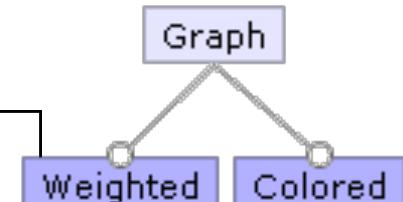
class Color {
    static void setDisplayColor(Color c) { ... }
}

```

```

class Weight { void print() { ... } }

```



# Mining Features from the Object-Oriented Source Code of Software Variants by Combining Lexical and Structural Similarity

R. AL-msie'deen, A.-D. Seriai, M. Huchard

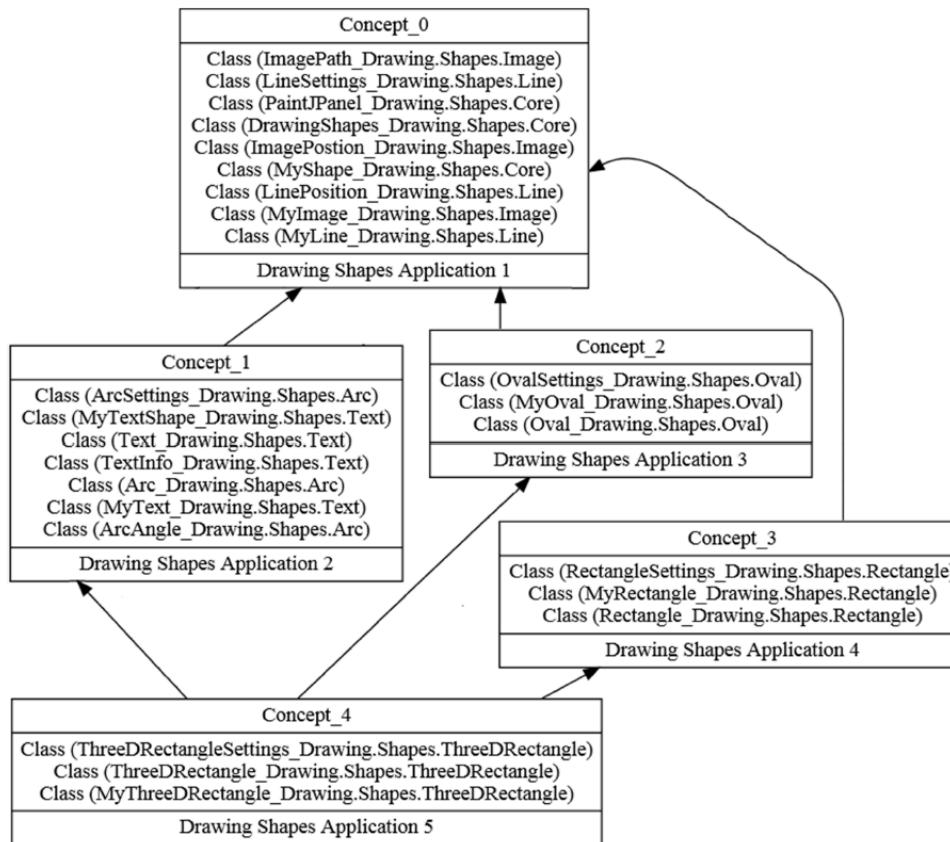
LIRMM / CNRS & Montpellier 2 University, Montpellier, France

Al-msiedee, Abdelhak.Seriai, huchard@lirmm.fr

C. Urtado and S. Vauttier

LGI2P / Ecole des Mines d'Alès, Nîmes, France

Christelle.Urtado, Sylvain.Vauttier@mines-ales.fr



	Class (ArcSettings_Drawing.Shapes.Arc)	Class (Arc_Drawing.Shapes.Arc)	Class (ArcAngle_Drawing.Shapes.Arc)	Class (MyTextShape_Drawing.Shapes.Text)	Class (Text_Drawing.Shapes.Text)	Class (TextInfo_Drawing.Shapes.Text)	Class (MyText_Drawing.Shapes.Text)
Class (ArcSettings_Drawing.Shapes.Arc)	×	×	×				
Class (Arc_Drawing.Shapes.Arc)	×	×	×				
Class (ArcAngle_Drawing.Shapes.Arc)	×	×	×				
Class (MyTextShape_Drawing.Shapes.Text)				×	×	×	×
Class (Text_Drawing.Shapes.Text)				×	×	×	×
Class (TextInfo_Drawing.Shapes.Text)				×	×	×	×
Class (MyText_Drawing.Shapes.Text)				×	×	×	×

# Video Encoder



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



(a) Variant #1 of video sequence



(b) Variant #2 of video sequence



(c) Variant #3 of video sequence



(d) Variant #4 of video sequence

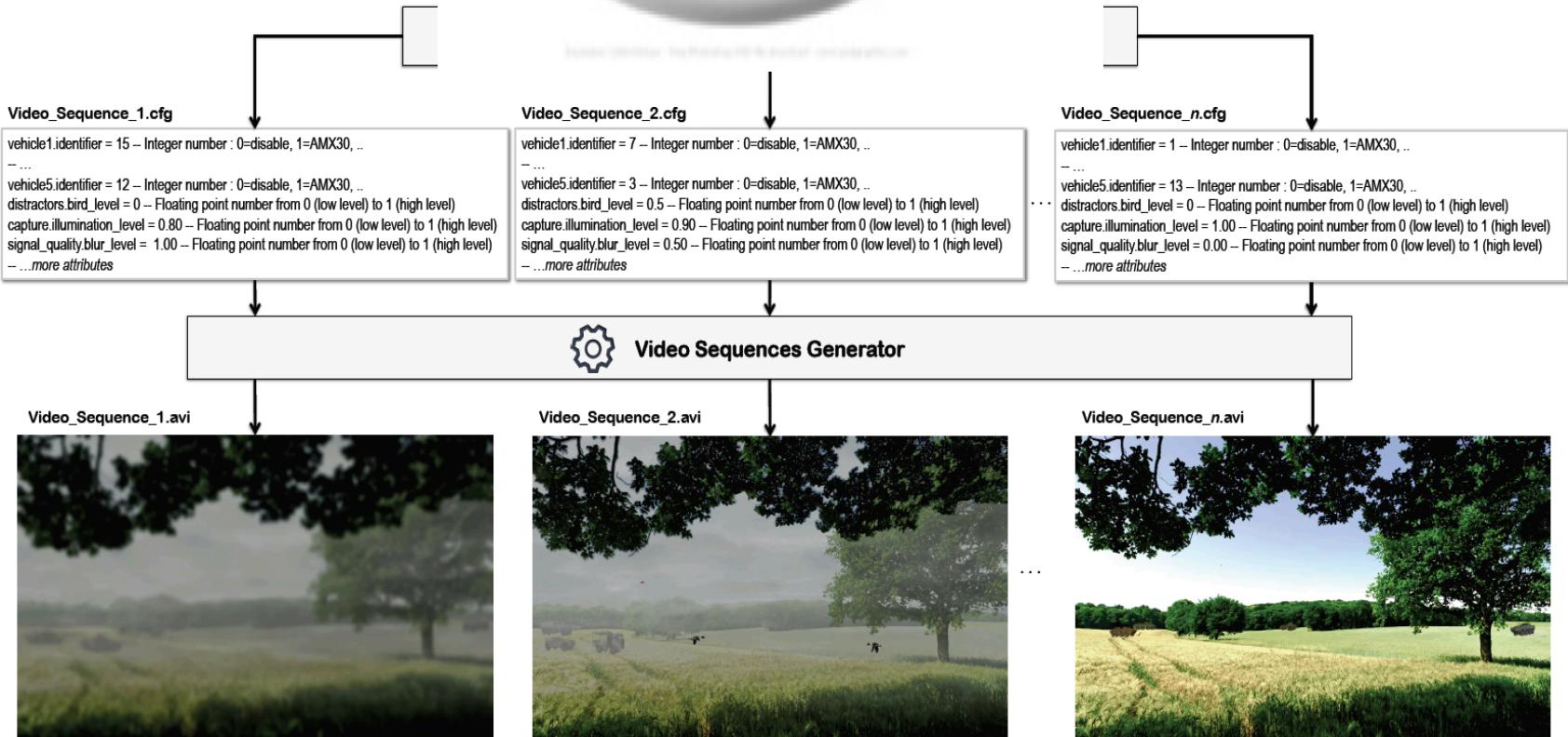


(e) Variant #5 of video sequence



(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]
```

1

```
//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]
```

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.

**Short** Short is for the Y direction.

**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

Made

Collections

Remixes

#### 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.

20865

2444

Found in Containers

Report Thing as Inappropriate

Makes

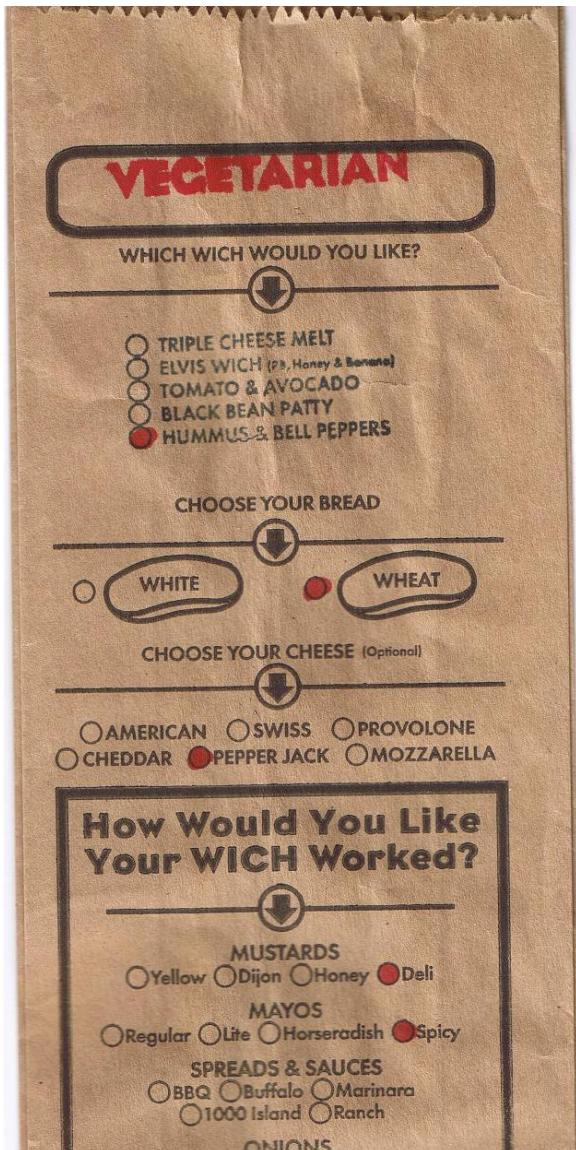
view more >

Customize body
Design key
Customize ruler
Printer platform se

▲
◀
▶
▼

(credits: Christian Kaestner's slide)

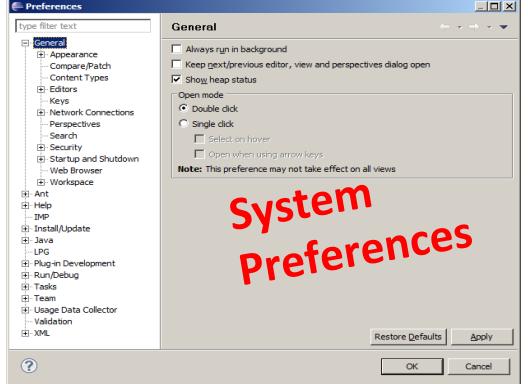
# Food? Product lines!



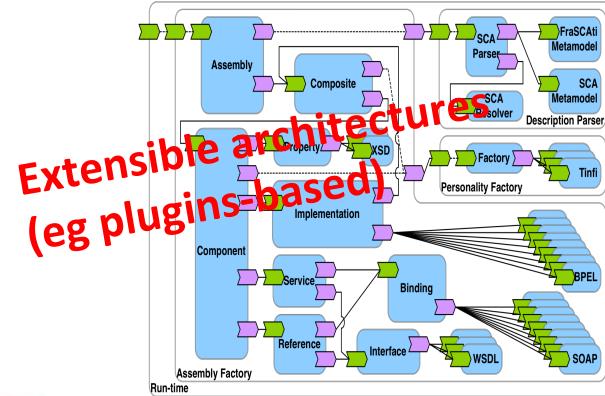


NEW KANGOO VAN RANGE

Configurators



System  
Preferences



Extensible architectures  
(eg plugins-based)



Comparison of \*

# Product Lines and Variability



[httpd.conf](http://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

UseSendfile Off

HostnameLookups Off

ErrorLog logs/error.log
LogLevel error

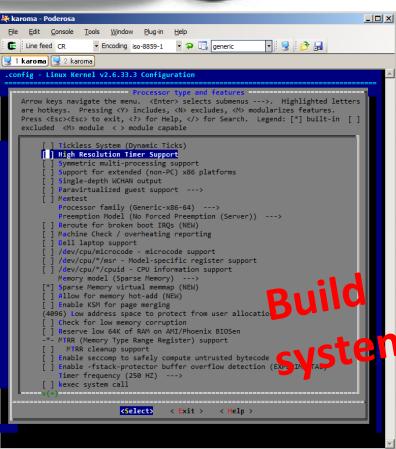
PidFile logs/httpd.pid

Timeout 300

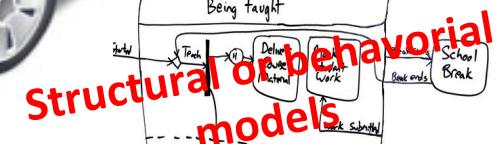
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

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

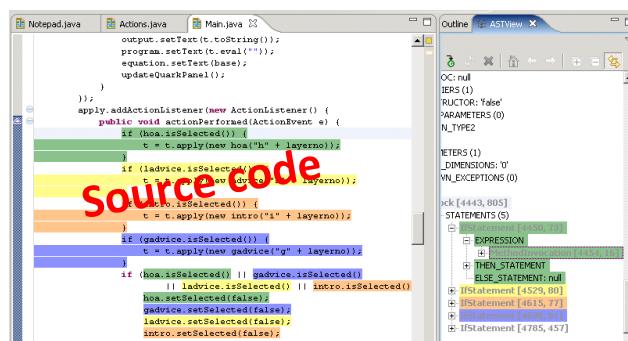
Configuration  
files



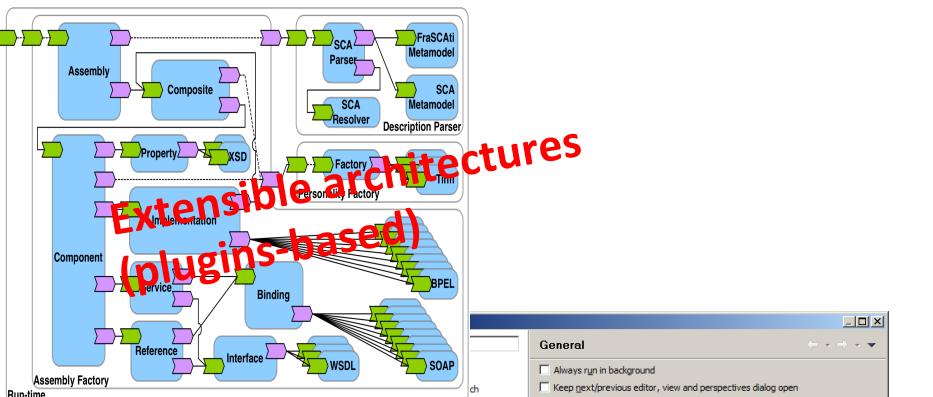
Build  
systems



Structural or behavioral  
models



Source code



<http://httpd.conf -- win32 App 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

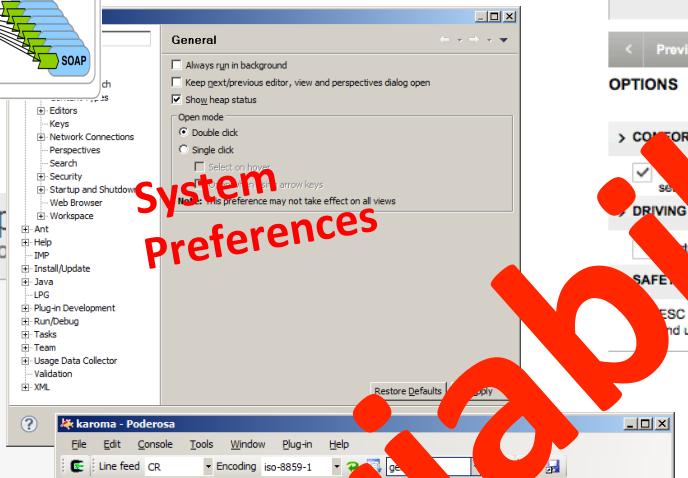
PidFile logs/httpd.pid

Timeout 300

KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

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

System Preferences



```

karoma - Poderosa
File Edit Console Tools Window Plugin Help
[ ] Line feed CR Encoding iso-8859-1
[ ] karoma

.config - Linux Kernel v2.6.33.3 Config
Processor type and features
Arrow keys navigate the menu system; submenus >>>. Highlighted letters are hotkeys. Pressing <Shift><Up> or <Shift><Down> modulates features. Press <Esc><F2> to exit <F2> to search. Legend: [*] built-in [ ] excluded [ ] module [ ] module

[ ] tick System (Dynamic Tick Support)
[ ] High Resolution Timer Support
[ ] Symmetric multi-processing support
[ ] Supports extended memory (x86 platforms)
[ ] Single channel WCHAN output
[ ] Multi-threaded guest support >>>

[*] tick Generic (x86-64) --->
[ ] Preemptible (No Forced Preemption (Server)) --->
[ ] Renroute: 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)
[ ] Intel KSM for page merging (4096)
[ ] Low address space to protect from user location
[ ] check for low memory for kernel
[ ] Reserve memory for kernel on NVidia Phoenix BIOSen
[*] TMR (Time Management Register) support
[ ] TMR (Time Management Register) support
[ ] enable seccomp to safely compute untrusted bytecode
[ ] enable -fstack-protector buffer overflow detection (EXPERIMENTAL)
[ ] Timer frequency (250 Hz) --->
[ ] exec system call

<Select> <Exit> <Help>
```

Build systems



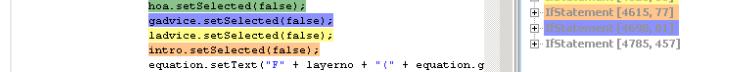
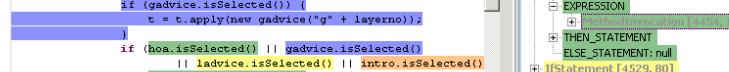
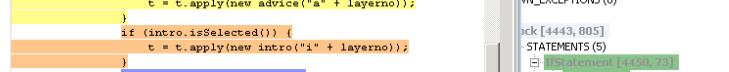
Compare Adobe Creative Suite 4 editions	
Design Premium	US\$1,799
Design Standard	US\$1,399
Web Premium	US\$1,699
Web Standard	US\$999
Production Premium	US\$1,699
Production Standard	US\$799
Master Collection	US\$2,499

Comparison of Product

```

Notepad.java Actions.java Main.java
output.setText(t.toString());
program.setText(t.eval(""));
equation.setText("a");
updateQuarkPanel();
}
apply.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
if (hoa.isSelected()) {
t = t.apply("a" + layerno);
}
if (ladvice.isSelected()) {
t = t.apply(new ladvice("a" + layerno));
}
if (intro.isSelected()) {
t = t.apply(new intro("a" + layerno));
}
if (gadvice.isSelected()) {
t = t.apply(new gadvice("a" + layerno));
}
if (hoa.isSelected() || gadvice.isSelected() ||
ladvice.isSelected() || intro.isSelected()) {
hoa.setSelected(false);
gadvice.setSelected(false);
ladvice.setSelected(false);
intro.setSelected(false);
equation.setText("F" + layerno + "(" + equation.get
+ ")");
}
}
}
lock[4443, 805]
STATEMENTS(5)
  + IFStatement[4443, 805]
    + EXPRESSION
      + THEN_STATEMENT
        + ELSE_STATEMENT: null
    + IFStatement[4529, 80]
    + IFStatement[4615, 77]
    + IFStatement[4693, 81]
    + IFStatement[4785, 457]
```

Source code



# Quizz Time

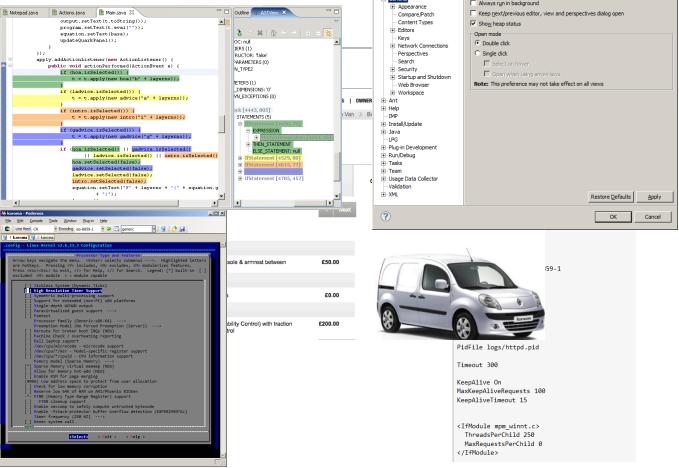
Give three examples of software product lines (also called configurable systems or variability-intensive systems)

# Software is eating the world (any company will be a software company)

If you have the super-power to  
“**vary**” (e.g., for delivering customized  
solutions to customers; or for  
engineering customized solutions for  
addressing your specific problems)

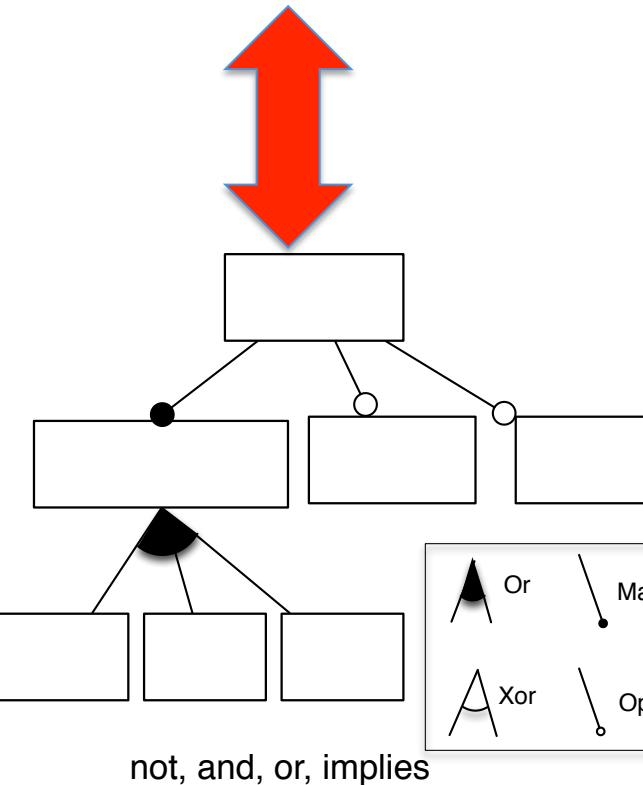
Then you will rule the world





Variants of code (e.g., Java ou C)  
 Variants of user interfaces  
 Variants of video sequences  
 Variants of models (e.g., UML or SysML)  
 Variants of « things » (3D models)

...



# Variability Models

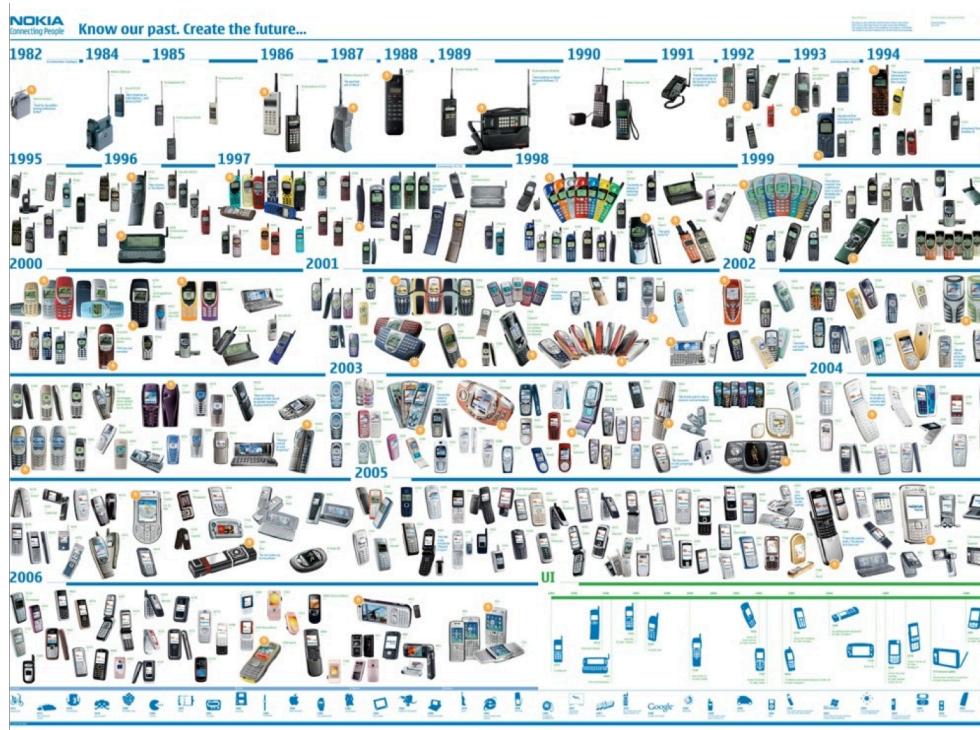
(feature models)

# Variability: two definitions

- “the **ability** of a software system or artifact to be efficiently extended, changed, customized or configured for use in a particular context” (Svahnberg et al. 2005)
  - software/**customization** perspective
- “an assumption about how members of a family may **differ** from each other” (Weiss and Lai 1999)
  - more related to the notions of **domain** and **commonality**

# Variability in time vs in space

- **Variability in Time (releases)**
  - the existence of different **versions** of an artifact that are valid at different times
- **Variability in Space (variants)**
  - the existence of an artifact in different **shapes** at the same time

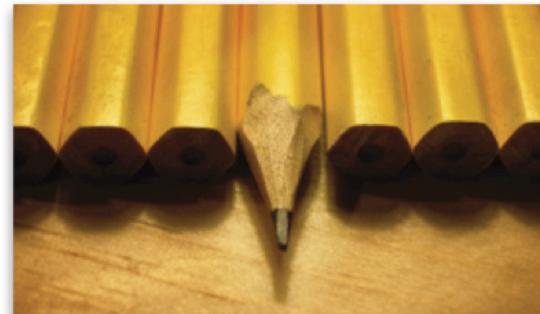


# Benefits

Improve product reliability

Improve usability

Improve consistency across products...



# Benefits

Reduce production costs



Reduce certification costs



Shorten time-to-market



# Hall of Fame

[splc.net/fame.html](http://splc.net/fame.html)



**BOSCH**

Invented for life



invent



**PHILIPS**



**NOKIA**  
Connecting People

**CelsiusTech**

**ERICSSON**



**Lucent Technologies**  
Bell Labs Innovations





# Printer Firmware

- Production cost reduced by 75%
- Development time reduced by 33%
- Reported defects reduced by 96%



A large, intricate 3D white maze is set against a light gray background. The maze consists of many interconnected paths and dead ends, creating a complex network of levels and corners. It occupies the entire frame, from the top left to the bottom right.

**Variability = Complexity**

(credits: Christian Kaestner's slide)

# 33 features

optional, independent



a unique variant for every  
person on this planet

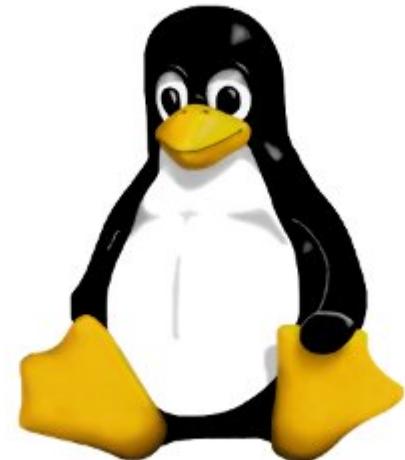
320<sup>optional, independent</sup>  
features

more variants than estimated  
atoms in the universe



2000 features

10000  
features



# Software product line and Variability engineering

Basic techniques

# What is new?

**Family vs single systems**

**Focus on reuse**

**Domain engineering**

**Factoring out commonality**

**Managing variability**

« variability »

Is it really new?

# Intentional Code Cloning

~ Copy & Paste

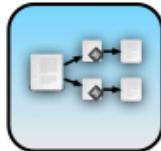
# Code Cloning (example, Linux driver)

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;  
}  
.....
```

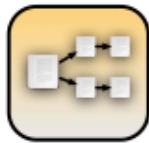


# Replicate & Specialize

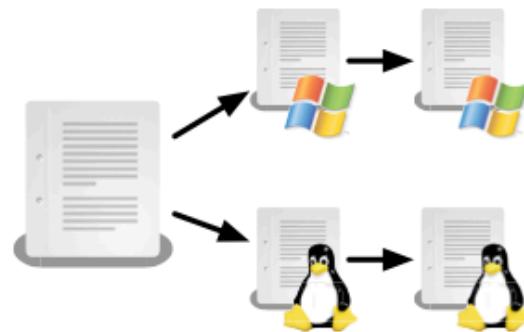


## **Clone to reuse and adapt existing solutions**

- + 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

# “Cloning Considered Harmful” Considered Harmful

Cory Kapser and Michael W. Godfrey  
Software Architecture Group (SWAG)

David R. Cheriton School of Computer Science, University of Waterloo  
{cjkapser, migod}@uwaterloo.ca

## "Cloning considered harmful" considered harmful: patterns of cloning in software

Authors: [Cory J. Kapser](#)

[Software Architecture Group \(SWAG\)](#) [David R. Cheriton](#)  
[School of Computer Science, University of Waterloo,](#)  
[Waterloo, Canada](#)

[Michael W. Godfrey](#)

[Software Architecture Group \(SWAG\)](#) [David R. Cheriton](#)  
[School of Computer Science, University of Waterloo,](#)  
[Waterloo, Canada](#)



2008 Article

Published in:

- Journal

[Empirical Software Engineering archive](#)

Volume 13 Issue 6, December 2008

Pages 645 - 692

Kluwer Academic Publishers Hingham, MA, USA

[table of contents](#) doi>[10.1007/s10664-008-9076-6](https://doi.org/10.1007/s10664-008-9076-6)

### [Bibliometrics](#)

- Downloads (6 Weeks): n/a
- Downloads (12 Months): n/a
- Downloads (cumulative): n/a
- Citation Count: 48

# Parameter

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\kaestner.INFORMATIK.000>dir /?
Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/A[[:l]attributes] [/B] [/C] [/D] [/L] [/N]
  [/O[[:l]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:l]timefield]] [/W] [/X] [/4]

[drive:][path][filename]
      Specifies drive, directory, and/or files to list.

/A          Displays files with specified attributes.
attributes   D  Directories                  R  Read-only files
              H  Hidden files                A  Files ready for archiving
              S  System files                I  Not content indexed files
              L  Reparse Points             -  Prefix meaning not
/B          Uses bare format (no heading information or summary).
/C          Display the thousand separator in file sizes. This is the
            default. Use /-C to disable display of separator.
/D          Same as wide but files are list sorted by column.
/L          Uses lowercase.
/N          New long list format where filenames are on the far right.
/O          List by files in sorted order.
sortorder    N  By name (alphabetic)        S  By size (smallest first)
              E  By extension (alphabetic)  D  By date/time (oldest first)
              G  Group directories first   -  Prefix to reverse order
/P          Pauses after each screenful of information.
```

# 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     }
}
```

# Command Line Options



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

# 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
```

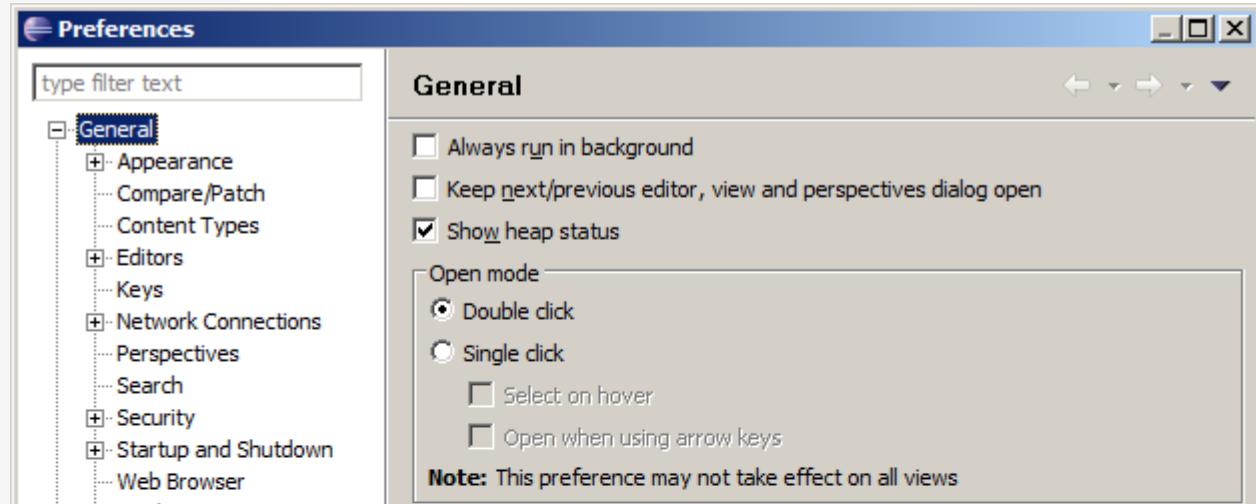
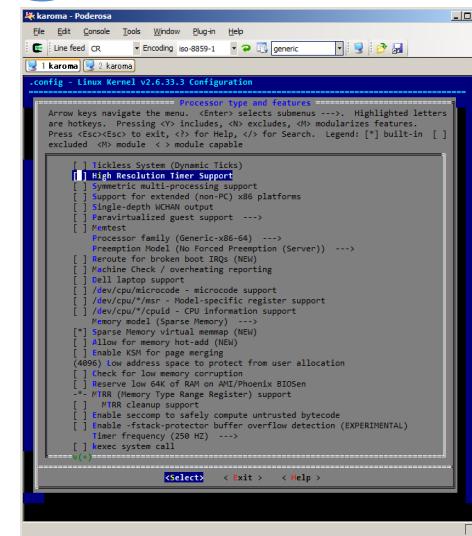
```
HostnameLookups Off
```

```
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
```

# 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

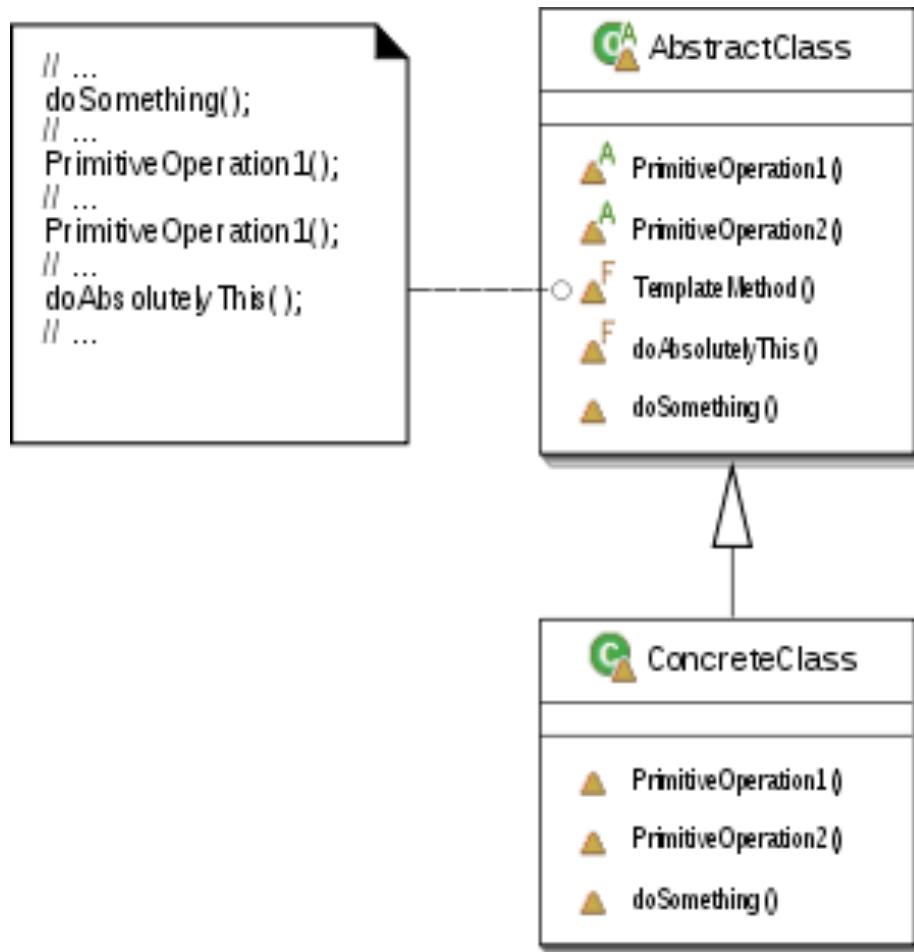
Factory

Strategy

Decorator

....

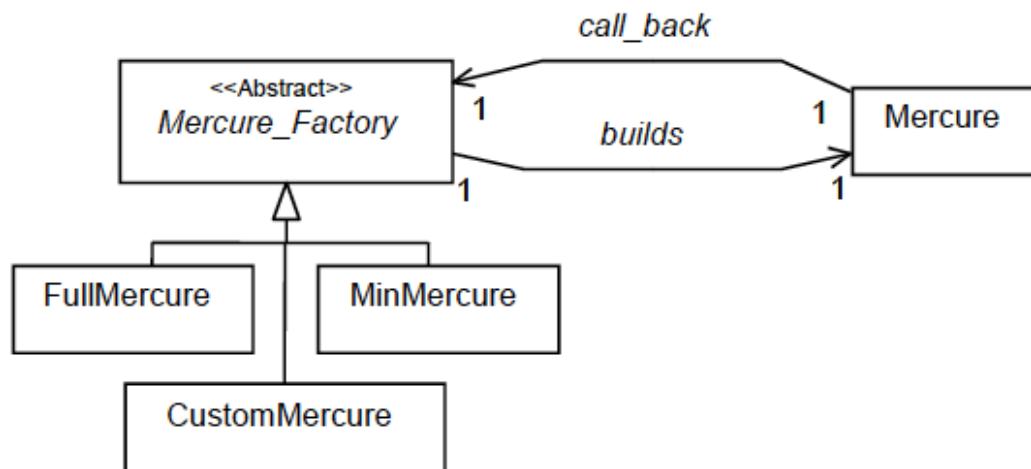
# Template Method



# 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

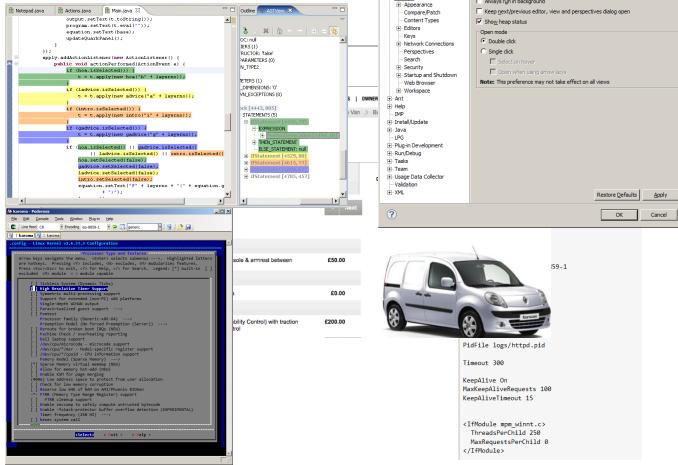
# Plugin-based systems

# (Active) Annotations

can have parameters

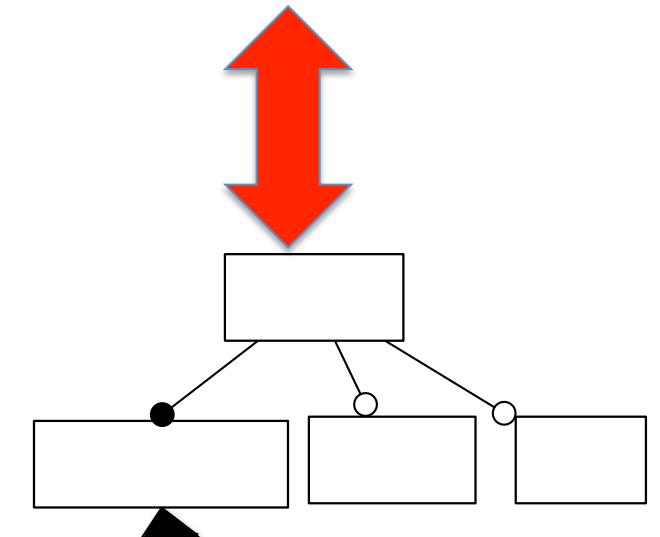
# Metamodeling and Domain-Specific Languages

# Variability (general approach)

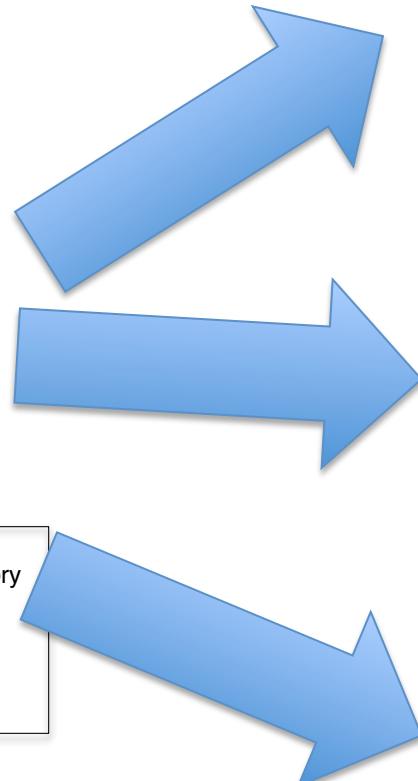


Variants of code (e.g., Java ou C)  
 Variants of user interfaces  
 Variants of video sequences  
 Variants of models (e.g., UML or SysML)

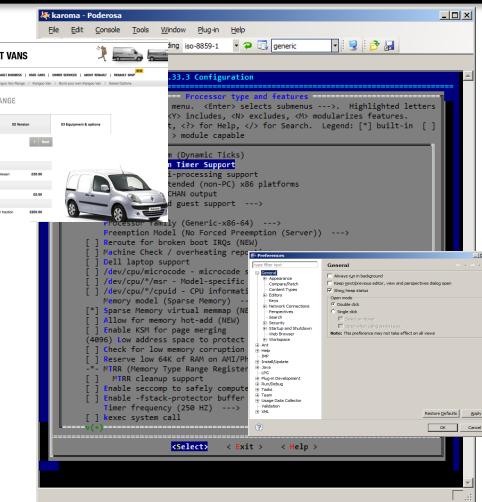
...



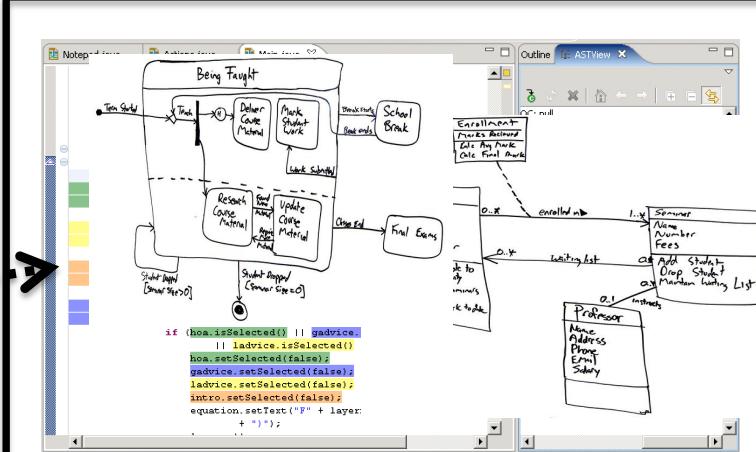
not, and, or, implies



# Variability Models (feature models)



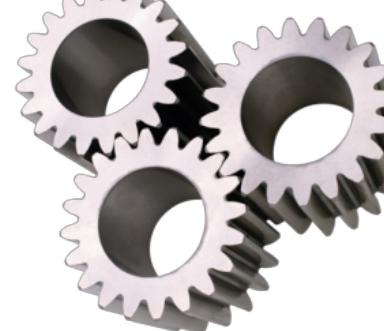
# Variability Model



## Base Artefacts (e.g., models)

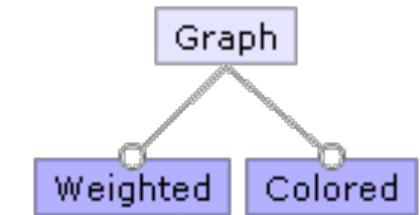


# Configuration



# Software Generator (derivation engine)





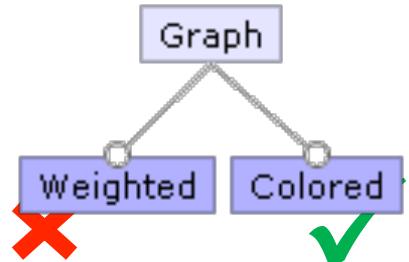
## Variability Model

mapping

```

class Graph {
    Vector nv = new Vector(); Vector ev = new Vector();
    Edge add(Node n, Node m) {
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
        /*if[WEIGHT]*/
        e.weight = new Weight();
        /*end[WEIGHT]*/
        return e;
    }
    /*if[WEIGHT]*/
    Edge add(Node n, Node m, Weight w) {
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
        e.weight = w; return e;
    }
    /*end[WEIGHT]*/
    void print() {
        for(int i = 0; i < ev.size(); i++) {
            ((Edge)ev.get(i)).print();
        }
    }
}
  
```

## Base Artefacts



```

class Graph {
    Vector nv = new Vector(); Vector ev = new Vector();
    Edge add(Node n, Node m) {
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
        return e;
    }
    void print() {
        for(int i = 0; i < ev.size(); i++) {
            ((Edge)ev.get(i)).print();
        }
    }
}
  
```

## Software Generator (derivation engine)

# Mapping: an example

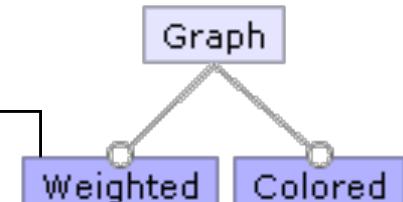
```
class Graph {  
    Vector nv = new Vector(); Vector ev = new Vector();  
    Edge add(Node n, Node m) {  
        Edge e = new Edge(n, m);  
        nv.add(n); nv.add(m); ev.add(e);  
        e.weight = new Weight();  
        return e;  
    }  
    Edge add(Node n, Node m, Weight w)  
    Edge e = new Edge(n, m);  
    nv.add(n); nv.add(m); ev.add(e);  
    e.weight = w; return e;  
}  
void print() {  
    for(int i = 0; i < ev.size(); i++) {  
        ((Edge)ev.get(i)).print();  
    }  
}
```

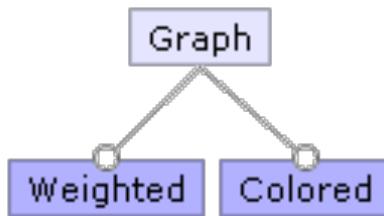
```
class Node {  
    int id = 0;  
    Color color = new Color();  
    void print() {  
        Color.setDisplayColor(color);  
        System.out.print(id);  
    }  
}
```

```
class Edge {  
    Node a, b;  
    Color color = new Color();  
    Weight weight = new Weight();  
    Edge(Node _a, Node _b) { a = _a; b = _b; }  
    void print() {  
        Color.setDisplayColor(color);  
        a.print(); b.print();  
        weight.print();  
    }  
}
```

```
class Color {  
    static void setDisplayColor(Color c) { ... }  
}
```

```
class Weight { void print() { ... } }
```





```

class Graph {
    Vector nv = new Vector(); Vector ev = new Vector();
    Edge add(Node n, Node m) {
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
        /*if[WEIGHT]*/
        e.weight = new Weight();
        /*end[WEIGHT]*/
        return e;
    }
    /*if[WEIGHT]*/
    Edge add(Node n, Node m, Weight w)
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
        e.weight = w; return e;
    }
    /*end[WEIGHT]*/
    void print() {
        for(int i = 0; i < ev.size(); i++) {
            ((Edge)ev.get(i)).print();
        }
    }
}

/*if[WEIGHT]*/
class Weight { void print() { ... } }
/*end[WEIGHT]*/

```

```

class Edge {
    Node a, b;
    /*if[COLOR]*/
    Color color = new Color();
    /*end[COLOR]*/
    /*if[WEIGHT]*/
    Weight weight;
    /*end[WEIGHT]*/
    Edge(Node _a, Node _b) { a = _a; b = _b; }
    void print() {
        /*if[COLOR]*/
        Color.setDisplayColor(color);
        /*end[COLOR]*/
        a.print(); b.print();
        /*if[WEIGHT]*/
        weight.print();
        /*end[WEIGHT]*/
    }
}

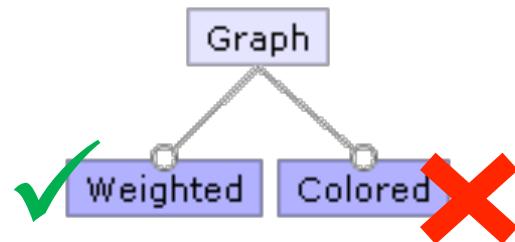
/*if[COLOR]*/
class Color {
    static void setDisplayColor(Color c) { ... }
}
/*end[COLOR]*/

```

```

class Node {
    int id = 0;
    /*if[COLOR]*/
}

```



```

class Graph {
    Vector nv = new Vector(); Vector ev = new Vector();
    Edge add(Node n, Node m) {
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
        e.weight = new Weight();
        return e;
    }
    Edge add(Node n, Node m, Weight w)
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
        e.weight = w; return e;
    }
    void print() {
        for(int i = 0; i < ev.size(); i++) {
            ((Edge)ev.get(i)).print();
        }
    }
}
    
```

```

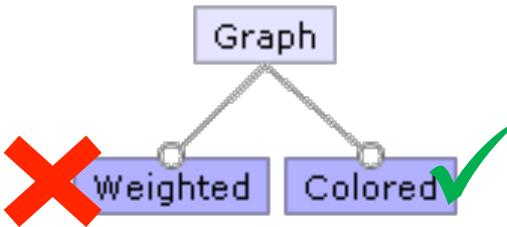
class Edge {
    Node a, b;
    Weight weight;
    Edge(Node _a, Node _b) { a = _a; b = _b; }
    void print() {
        a.print(); b.print();
        weight.print();
    }
}
    
```

```

class Node {
    int id = 0;
    void print() {
        System.out.print(id);
    }
}
    
```

```

class Weight { void print() { ... } }
    
```



```

class Graph {
    Vector nv = new Vector(); Vector ev = new Vector();
    Edge add(Node n, Node m) {
        Edge e = new Edge(n, m);
        nv.add(n); nv.add(m); ev.add(e);
    return e;
    }
    void print() {
        for(int i = 0; i < ev.size(); i++) {
            ((Edge)ev.get(i)).print();
        }
    }
}

```

```

class Edge {
    Node a, b;
    Color color = new Color();
    Edge(Node _a, Node _b) { a = _a; b = _b; }
    void print() {
        Color.setDisplayColor(color);
        a.print(); b.print();
    }
}

```

```

class Color {
    static void setDisplayColor(Color c) { ... }
}

```

```

class Node {
    int id = 0;
    Color color = new Color();
    void print() {
        Color.setDisplayColor(color);
        System.out.print(id);
    }
}

```

# Kästner “Virtual Separation of Concerns: Toward Preprocessors 2.0”

```
1 class Add extends Expr { //yellow
2     Expr left, right;
3     Add(Expr l, Expr r)
4         { left=l; right=r; }
5     double eval() { //red
6         return left.eval() +
7             right.eval();
8     }
9     void print() { //blue
10        left.print();
11        System.out.print("+");
12        right.print();
13    }
14 }
```

Features: ADD, EVAL, PRINT

The screenshot shows a Java development environment with two tabs open: `Notepad.java` and `Main.java`. The `Main.java` tab is active, displaying the following code:

```
        output.setText(t.toString());
        program.setText(t.eval(""));
        equation.setText(base);
        updateQuarkPanel();
    }
});
apply.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (hoa.isSelected()) {
            t = t.apply(new hoa("h" + layerno));
        }
        if (ladvice.isSelected()) {
            t = t.apply(new advice("a" + layerno));
        }
        if (intro.isSelected()) {
            t = t.apply(new intro("i" + layerno));
        }
        if (gadvice.isSelected()) {
            t = t.apply(new gadvice("g" + layerno));
        }
        if (hoa.isSelected() || gadvice.isSelected()
            || ladvice.isSelected() || intro.isSelected())
            hoa.setSelected(false);
        gadvice.setSelected(false);
        ladvice.setSelected(false);
        intro.setSelected(false);
        equation.setText("F" + layerno + "(" + equation.g
            + ")");
    }
});
```

The code uses color-coded syntax highlighting. The `ASTView` window on the right displays the Abstract Syntax Tree (AST) for the selected code. The tree structure is as follows:

- OC: null
- IERS (1)
  - RUCTOR: 'false'
- PARAMETERS (0)
- N\_TYPE2
- ETERS (1)
  - DIMENSIONS: '0'
- WN\_EXCEPTIONS (0)
- ock [4443, 805]
  - STATEMENTS (5)
    - IfStatement [4450, 73]
      - EXPRESSION
        - MethodInvocation [4454, 16]
      - THEN\_STATEMENT
      - ELSE\_STATEMENT: null
    - IfStatement [4529, 80]
    - IfStatement [4615, 77]
    - IfStatement [4698, 81]
    - IfStatement [4785, 457]

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

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

# JHIPSTER STACKER FOR JAVA EDIENS

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

## Base Artefacts

## Software Generator (derivation engine)

Branch: master

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

dubois 2 days ago Use Spring Boot's configuration meta-data  
9 contributors

184 lines (165 sloc) 9.69 KB

```
1 package org.jhipster.core.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 com.mongodb.Mongo;
8 import com.mongodb.MongoClient;
9 import com.mongodb.MongoCredential;
10 import org.slf4j.Logger;
11 import org.slf4j.LoggerFactory;
12 import org.springframework.beans.factory.annotation.Autowired;
13 import org.springframework.boot.autoconfigure.condition.ConditionalOnExpression;
14 import org.springframework.boot.autoconfigure.condition.ConditionalOnMissingBean;
15 import org.springframework.boot.autoconfigure.cache.CacheManager;
16 import org.springframework.boot.autoconfigure.mongo.MongoAutoConfiguration;
17 import org.springframework.boot.autoconfigure.mongo.MongoProperties;
18 import org.springframework.boot.autoconfigure.mongo.MongoDataSourceProperties;
19 import org.springframework.boot.autoconfigure.mongo.MongoDatabaseProperties;
20 import org.springframework.boot.autoconfigure.mongo.MongoEventListener;
21 import org.springframework.boot.autoconfigure.mongo.MongoRepositoryException;
22 import org.springframework.boot.autoconfigure.context.annotation.Bean;
23 import org.springframework.boot.autoconfigure.context.annotation.Configuration;
24 import org.springframework.boot.autoconfigure.context.annotation.Profile;
25 import org.springframework.boot.autoconfigure.context.annotation.Import;
26 import org.springframework.boot.autoconfigure.context.annotation.EnableConfiguration;
27 import org.springframework.core.convert.converter.Converter;
28 import org.springframework.core.io.ClassPathResource;
29 import org.springframework.data.elasticsearch.repository.config.EnableElasticsearchRepositories;
30 import org.springframework.data.mongodb.config.AbstractMongoConfiguration;
31 import org.springframework.data.mongodb.config.EnableMongoAuditing;
32 import org.springframework.data.mongodb.core.mapping.event.ValidatingMongoEventListener;
33 import org.springframework.data.mongodb.repository.config.EnableMongoRepositories;
34 import org.springframework.validation.beanvalidation.LocalValidatorFactoryBean;
35 import org.springframework.validation.beanvalidation.LocalValidatorFactoryBean;
```

Raw Blame History

edit

diff

blt

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

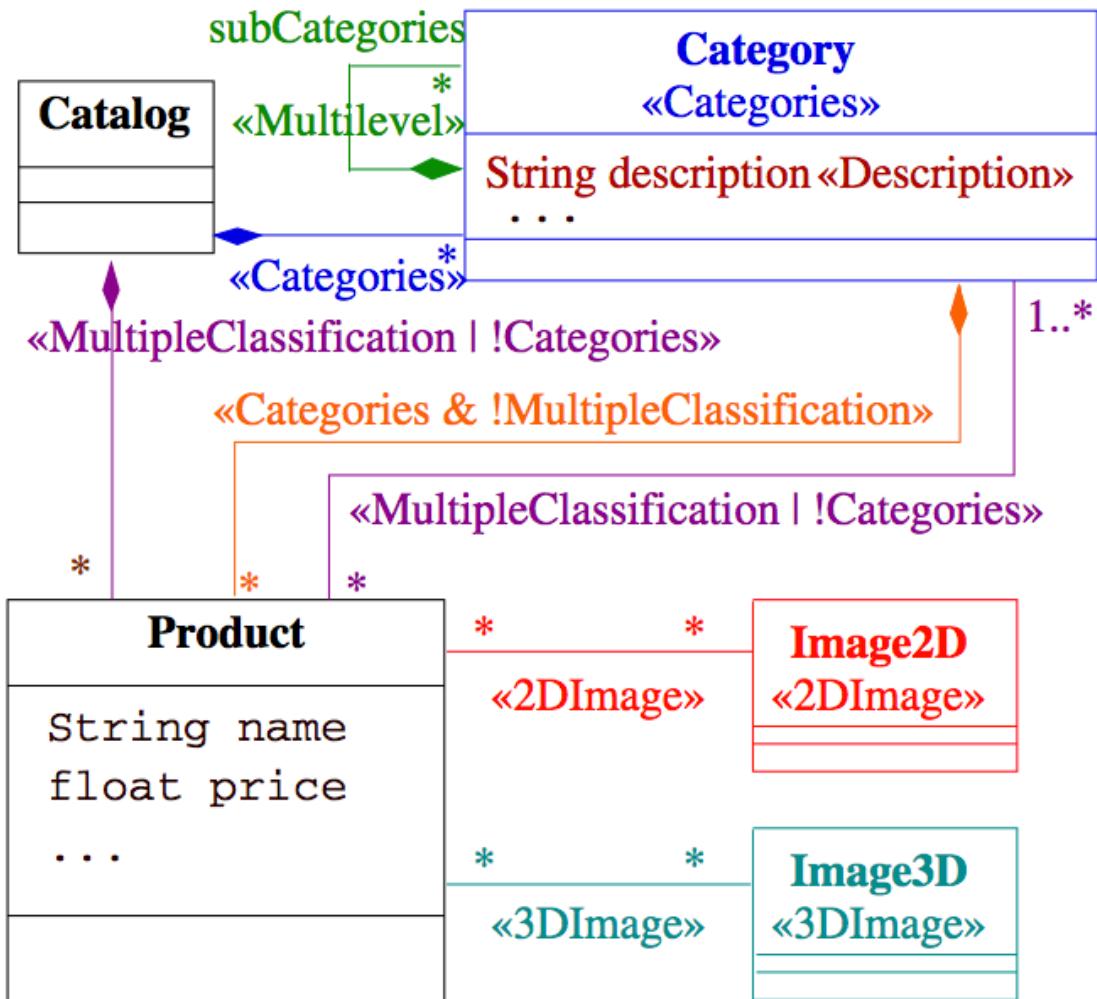
184 lines (165 sloc) | 9.69 KB

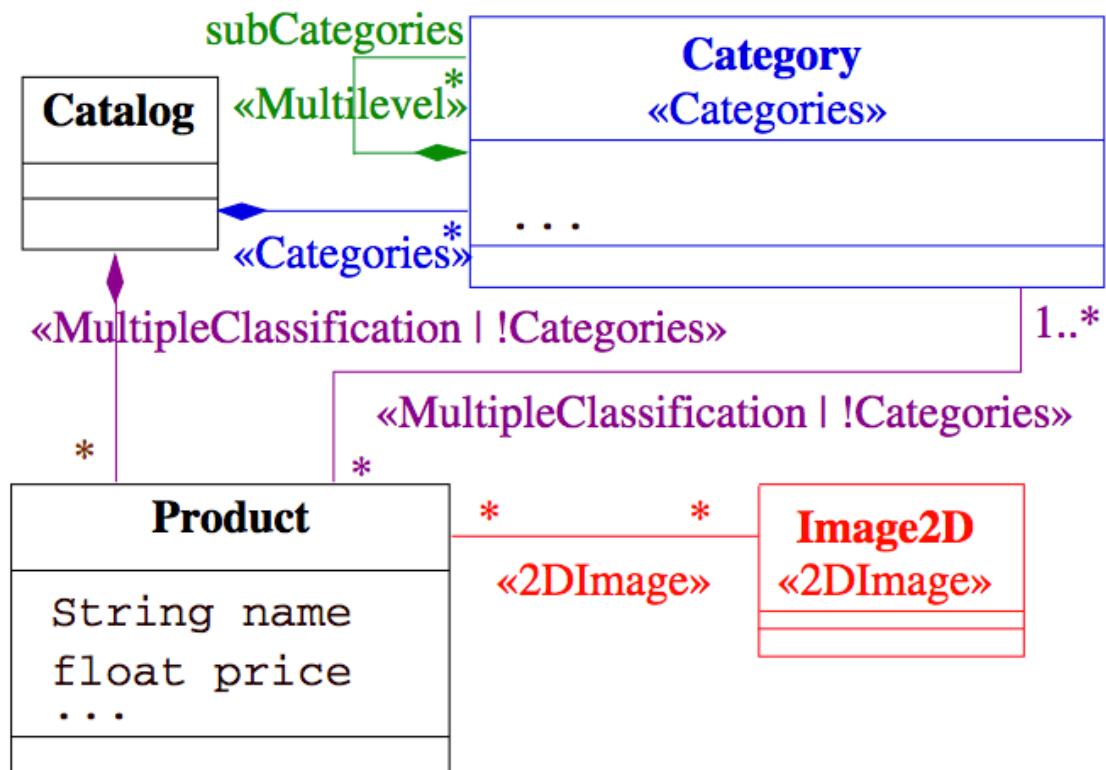
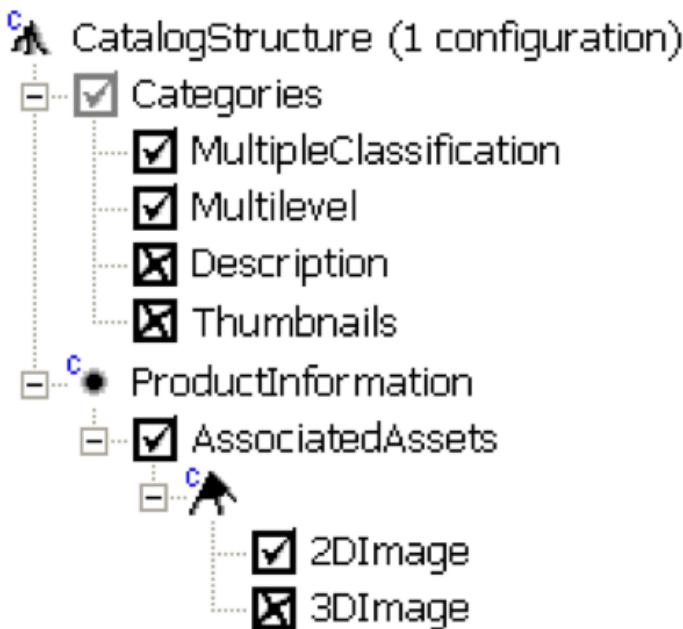
[Raw](#) [Blame](#) [History](#)   

```
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') { %>
```

## ▲ CatalogStructure (52 configurations)

- Categories
  - MultipleClassification
  - Multilevel
  - Description
  - Thumbnails
- ProductInformation
  - AssociatedAssets
    - 2DImage
    - 3DImage





# Variability in the Video Domain (first example)

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



```
1 /*! Wildmoka - v0.0.0 - 2014-11-04
2 * http://wildmoka.com/
3 * Copyright (c) 2014 ; Licensed *
4 var requirejs=require,define;!function()
5 return("input":c||"button"===c)&
6 if(h[0]=a,a.delegateTarget=this,!j)
7 g=fb.trim(d),c.className=g&&(c.c
8 if(k.length){var r=k.data("and")
9 },_mouseStart:function(){return!0
10 },show:function(){this._flashObj.s
11 d.append(e);var f=1,g=[];a.each(b,
```

## **ETAPE 2 : CHOISIS 3 BONS SOUVENIRS**



bref-service.cloudapp.net/pintvservices/getEp?a1=RANDOM&a2=RANDOM&a3=RANDOM&

← → ⌂ bref-service.cloudapp.net/pintvservices/getEp?a1=RANDOM&a2=RANDOM&a3=RANDOM&

"sq": ["dwlcjv", "1y60t3z", "1lyfhk", "wqzv0y",  
"1xxivi2", "1oxnvtu", "lolbe9", "wvo06o", "1u6y5t2",  
"1eqb8bw", "1j9aij7", "nr7jom", "1jmvt1y",  
"1qgn9dh", "1bv7rka", "19ykyyw", "5znrg7",  
"116hv1k"]

← → ⌂ bref30cdn.wildmoka.com/vidv2/116hv1k\_med.txt

```
#EXTINF:06.40,
http://bref30cdn.wildmoka.com/vidv2/116hv1k_med0.ts
#EXTINF:03.96,
http://bref30cdn.wildmoka.com/vidv2/116hv1k_med1.ts
#EXTINF:04.52,
http://bref30cdn.wildmoka.com/vidv2/116hv1k_med2.ts
#EXTINF:03.08,
http://bref30cdn.wildmoka.com/vidv2/116hv1k_med3.ts
```

```
}

1659055
: [ DESCHIENS, DECAUNESGARCIA, BURGER]
: "DESCHIENS"
: "DECAUNESGARCIA"
: "BURGER"
:e24=wxjqk2&e18=xi82vf&e05=x2nief&e21=cr3hzu&e25=1bv7rka&e06=loxnvtu&e04=ppdjpp&e12=1pt9ic1&e28=
"OK"
: [ml7ila, ly60t3z, 12eo6f2, ppdjpp, x2nief, loxnvtu, a7kxth, 1pt9ic1, nej6g3, 1hhuj2v, 1j9aij7,
: "ml7ila"
: "ly60t3z"
: "12eo6f2"
: "ppdjpp"
: "x2nief"
: "loxnvtu"
: "a7kxth"
: "1pt9ic1"
: "nej6g3"
: "1hhuj2v"
0: "1j9aij7"
1: "xi82vf"
2: "cr3hzu"
3: "wxjqk2"
4: "1bv7rka"
5: "5jjnew"
6: "f353uf"
7: "116hv1k"
```

- wget + curl = 363281 episodes
- 18 sequences, 400 « .ts »
- 63 alternatives for the 1st sequence
- And so on...

9    34    15    25    51    30    2    6    6    12    21    28    6    86    4    1

.DS\_Store

blank.ts

jg0meq\_medi0.ts

vp1

vp2

vp3

vp4

vp5

vp6

vp7

vp8

vp9

vp10

vp11

vp12

vp13

vp14

vp15

vp16

vp17

vp18

lu6y5t2\_medi0.ts

lu6y5t2\_medi1.ts

lu6y5t2\_medi2.ts

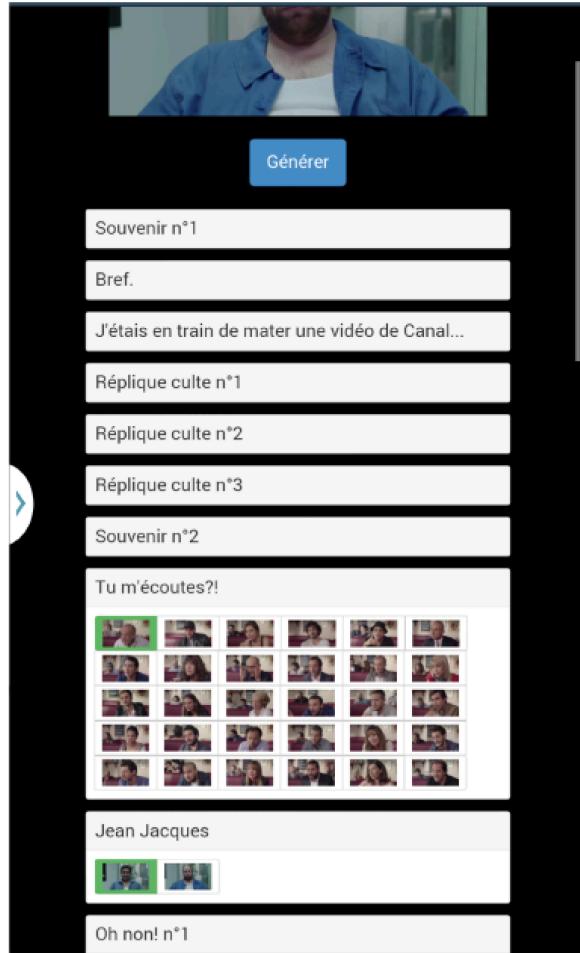
lu6y5t2\_medi3.ts

p8ocev\_medi0.ts

p8ocev\_medi1.ts

p8ocev\_medi2.ts





```
"sq": ["dwlcjv", "1y60t3z", "1lyfhk", "wqzv0y",
"1xxivi2", "1oxnvtu", "lolbe9", "wvo06o",
"1u6y5t2", "1eqb8bw", "1j9aij7", "nr7jom",
"1jmvi1y", "1qgn9dh", "1bv7rka", "19ykyyw",
"5znrg7", "116hv1k"]
```

← → C bref30cdn.wildmoka.com/vidv2/116hv1k\_med.txt

```
#EXTINF:06.40,
http://bref30cdn.wildmoka.com/vidv2/116hv1k_med0.ts
#EXTINF:03.96,
http://bref30cdn.wildmoka.com/vidv2/116hv1k_med1.ts
#EXTINF:04.52,
http://bref30cdn.wildmoka.com/vidv2/116hv1k_med2.ts
#EXTINF:03.08,
http://bref30cdn.wildmoka.com/vidv2/116hv1k_med3.ts
```

.DS\_Store  
blank.ts  
jg0meq\_med0.ts  
vp1  
vp2  
vp3  
vp4  
vp5  
vp6  
vp7  
vp8  
vp9  
vp10  
vp11  
vp12  
vp13  
vp14  
vp15  
vp16  
vp17  
vp18

1u6y5t2\_med0.ts  
1u6y5t2\_med1.ts  
1u6y5t2\_med2.ts  
1u6y5t2\_med3.ts  
p8ocev\_med0.ts  
p8ocev\_med1.ts  
p8ocev\_med2.ts

Figure 2: Re-engineering of the novel configurator (excerpt): users can now select a specific video for the 18 variation points identified during the reverse engineering of the original generator

# Variability in the Video Domain (second example)



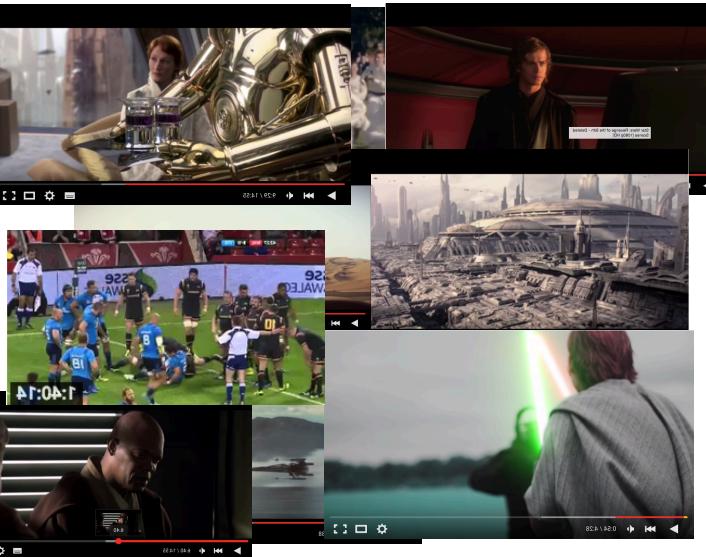


**Generator**  
~ composition of  
video sequences



**video  
variants**





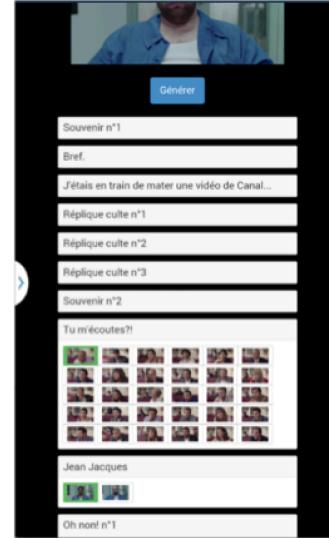
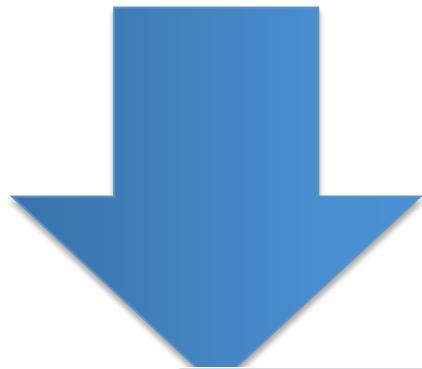
```

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"

```



- ## 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.videoogen

```
mandatory videooseq v1 "https://www.youtube.com/watch?v=PJNi1uYhV5w"  
optional videooseq v2 "v2folder/v2.mp4"  
alternatives v3 {  
    videooseq v31 "v3/seq1.mp4"  
    videooseq v32 "v3/seq1.mp4"  
    videooseq v33 "v3/seq1.mp4"  
}  
  
alternatives v4 {  
    videooseq v41 "v4/seq1.mp4"  
    videooseq v42 "v4/seq1.mp4"  
}  
mandatory videooseq 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?

# Variability in the Video Domain (third example)



# What are the differences?

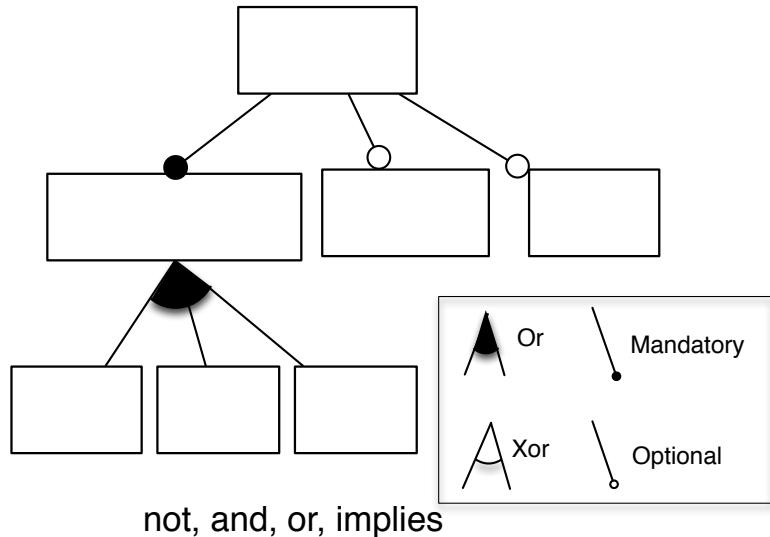




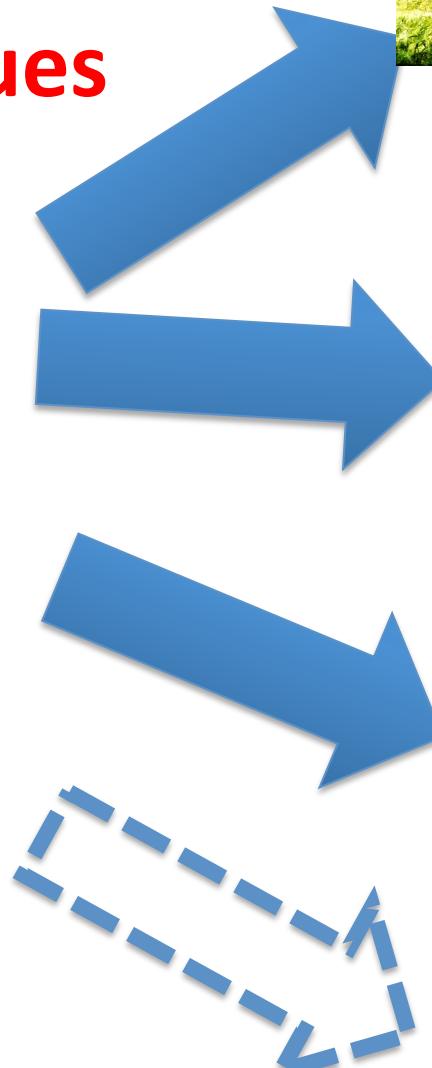
**aka what is the variability of a video?**



# We synthesize video sequence variants with variability techniques



## Variability Models (feature models)



# Why?

**Industrial needs:  
consumer and provider of  
video algorithms have severe**

**difficulties to **test** their  
algorithms on different kinds  
of inputs.**



# Test algorithms on different kinds of inputs that influence execution time, precision, and/or recall



**Algorithm 1**

0.63

0.81

0.43

0.39

**Algorithm 2**

0.93

0.92

0.3

0.03

**Algorithm 3**

0.82

0.81

0.8

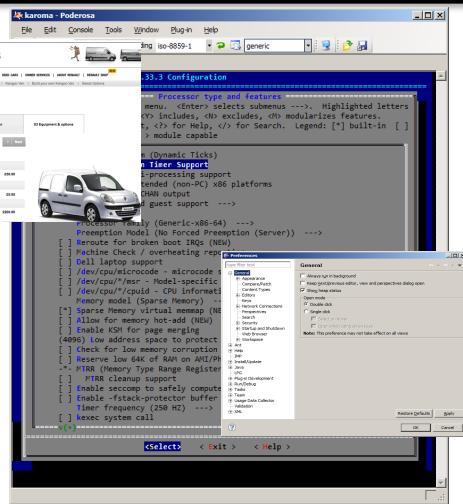
0.01

# Why?

**Problem: collecting videos is a key economic problem.**

**Solution: hundreds of video sequences with different characteristics.**





## Feature Model

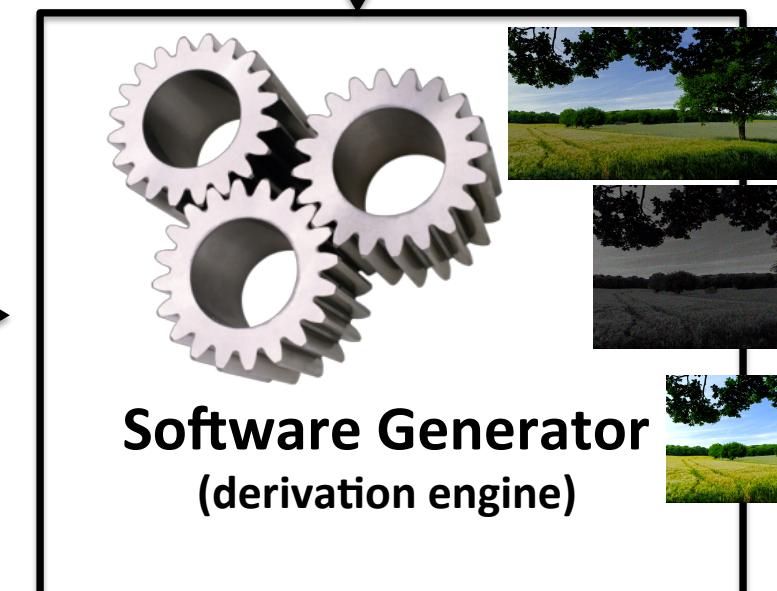
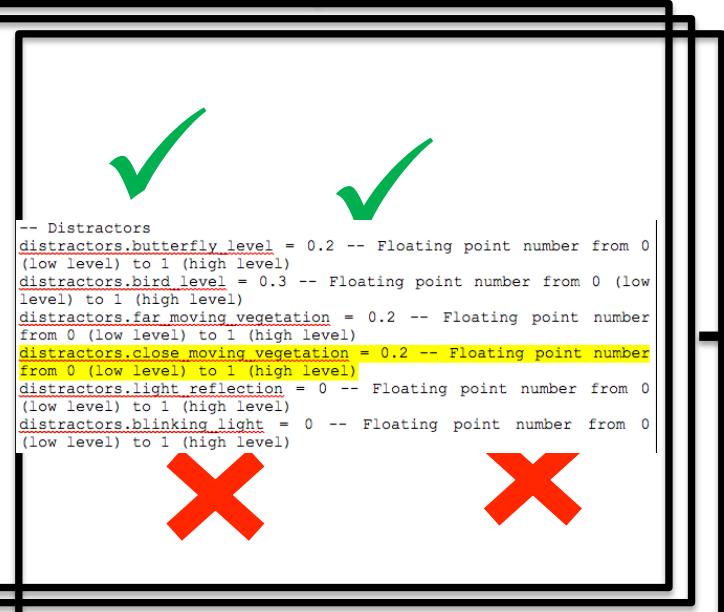
mapping

```

print("->Step9")
if(CFG.distractors.close_moving_vegetation~=0) then
    windvect5, precwindvect5, newwindvect5 =
    generate_wind_vector_field2(workwidth, workheight, 256, 1, 1, 35, picnum,
    precwindvect5, newwindvect5)
    windvectmul =
    windvect5:mul(24*CFG.distractors.close_moving_vegetation)
    globalvect = compose_vect(masque_feuilles_sombres, windvectmul, globalvect)
    hfvx, hfvy =
    windvect5:mul(6*CFG.distractors.close_moving_vegetation):to_matrix()
    hfvx = MATRIX.multerm(hfvx, invdepthmat)
    hfvy = MATRIX.multerm(hfvy, invdepthmat)
    lfvect = windvect2:resize_bilinear(windvect2.Width, windvect2.Height/16)
    lfvect = lfvect:resize_bilinear(windvect2.Width, windvect2.Height)
    lfvx, lfvy = lfvect:mul(
12*CFG.distractors.close_moving_vegetation):to_matrix()
    lfvx = MATRIX.multerm(lfvx, depthmat)
    lfvy = MATRIX.multerm(lfyv, depthmat*0.1)
    windvectcomp = VECT2D.new_from_matrices(MATRIX.addterm(lfvx, hfvx),
    MATRIX.addterm(lfvx, hfvy))
    globalvect = compose_vect(masque_ble, windvectcomp, globalvect)
    globalvect = compose_vect(masque_orge, windvectcomp, globalvect)
    globalvect = compose_vect(masque_ble_fond, windvectcomp, globalvect)
end

print("->Step10")

```



## Old Approach



Developers

*modify  
 $N$  times*

$N$  Configuration Files



Video Sequences Generator

*generates*



$N$  Video  
Sequences



## VM Approach

Developers and  
Domain Experts



*generates*

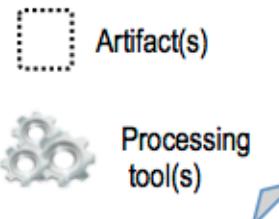


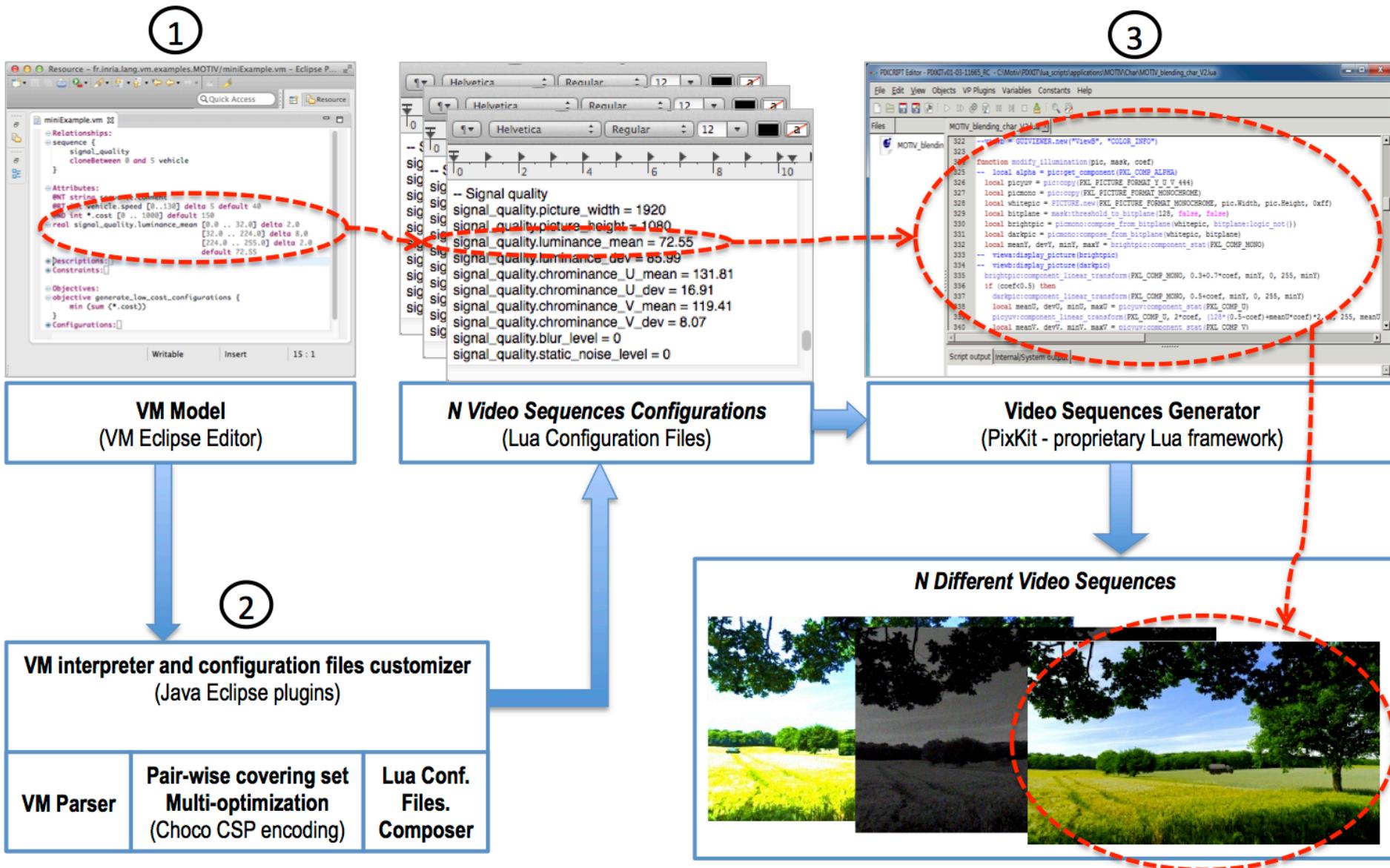
VM interpreter, and  
configuration files  
customizer

*model  
1 time*



VM model





# (configuration file)

```
-- Distractors
distractors.butterfly_level = 0.2 -- Floating point number from 0
(low level) to 1 (high level)
distractors.bird_level = 0.3 -- Floating point number from 0 (low
level) to 1 (high level)
distractors.far_moving_vegetation = 0.2 -- Floating point number
from 0 (low level) to 1 (high level)
distractors.close_moving_vegetation = 0.2 -- Floating point number
from 0 (low level) to 1 (high level)
distractors.light_reflection = 0 -- Floating point number from 0
(low level) to 1 (high level)
distractors.blinking_light = 0 -- Floating point number from 0
(low level) to 1 (high level)
```

---

```
print(">Step9")
if (CFG.distractors.close_moving_vegetation~=0) then
    windvect5, precwindvect5, newwindvect5 =
generate_wind_vector_field2(workwidth, workheight, 256, 1, 1, 35, picnum,
precwindvect5, newwindvect5)
    windvectmul =
windvect5:mul(24*CFG.distractors.close_moving_vegetation)
    globalvect = compose_vect(masque_feuilles_sombres, windvectmul, globalvect
hfvx, hfvy =
windvect0:mul(6*CFG.distractors.close_moving_vegetation):to_matrix()
    hfvx = MATRIX.multerm(hfvx, invdepthmat)
    hfvy = MATRIX.multerm(hfvy, invdepthmat)
    lfvect = windvect2:resize_bilinear(windvect2.Width, windvect2.Height/16)
    lfvect = lfvect:resize_bilinear(windvect2.Width, windvect2.Height)
    lfvx, lfvy = lfvect:mul(
12*CFG.distractors.close_moving_vegetation):to_matrix()
    lfvx = MATRIX.multerm(lfvx, depthmat)
    lfvy = MATRIX.multerm(lfvy, depthmat*0.1)
    windvectcomp = VECT2D.new_from_matrices(MATRIX.addterm(lfvx, hfvx),
MATRIX.addterm(lfvy, hfvy))
    globalvect = compose_vect(masque_ble, windvectcomp, globalvect)
    globalvect = compose_vect(masque_orge, windvectcomp, globalvect)
    globalvect = compose_vect(masque_ble_fond, windvectcomp, globalvect)
end
print(">Step10")
```

---

(Lua code)

Defects detection

Benchmarking

Incremental design

Performance prediction



**Algorithm 1**

0.63

0.81

0.43

0.39

**Algorithm 2**

0.93

0.92

0.3

0.03

**Algorithm 3**

0.82

0.81

0.8

0.01

# 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