FAMiliAR

Model-based Variability Engineering

Dr. Mathieu Acher
Associate Professor
http://www.mathieuacher.com

informatiques mathématiques Inria

UNIVERSITÉ DE RENNES 1

UMR IRISA

Course given for Johannes Kepler Universität (JKU) in Linz

(9th april 2014)

Variability and Software Product Lines

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

The three ways to build a (software) product

Indepently

„Clone & Own“

„Shared“ (reusable) Assets

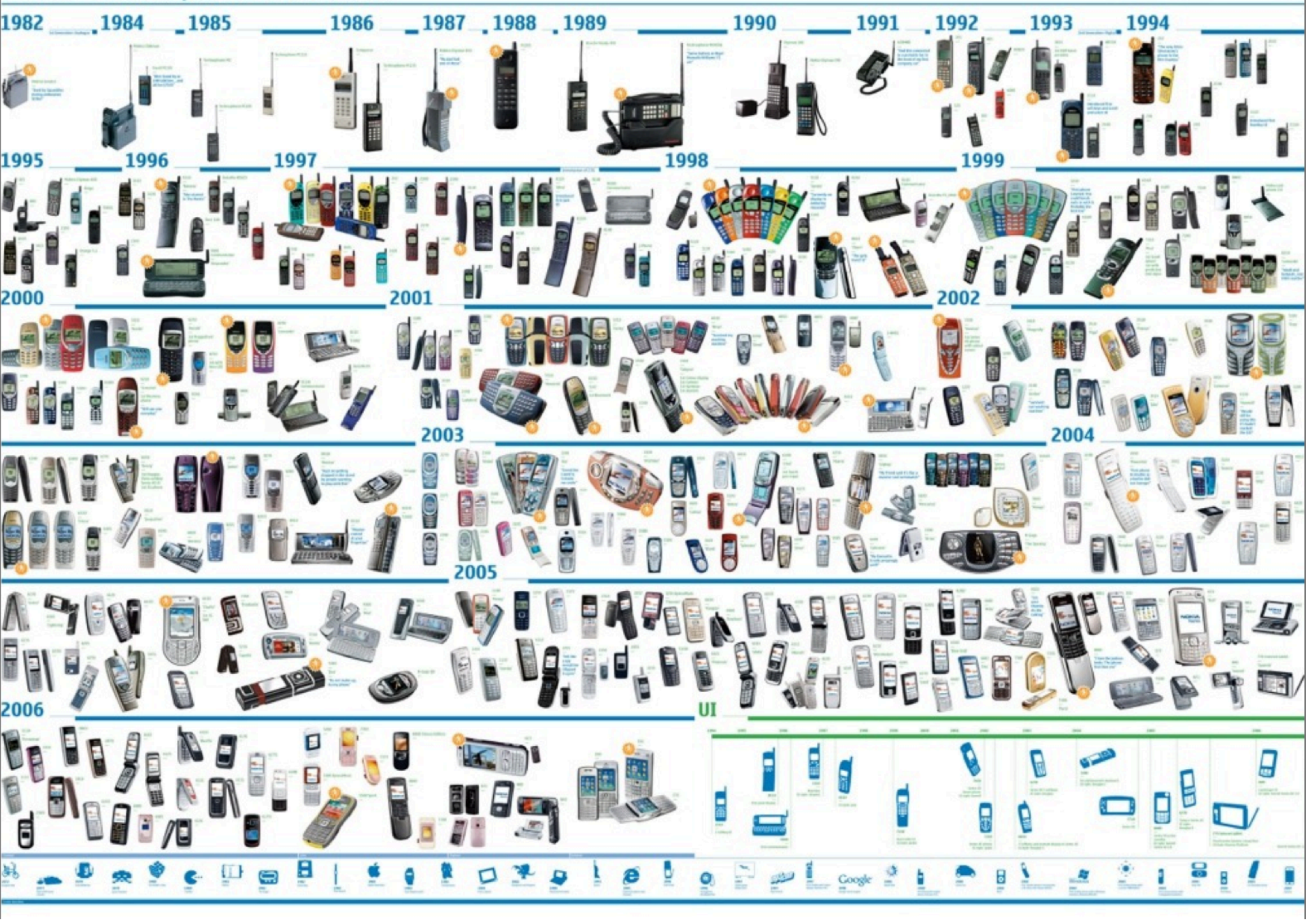
Software Product Lines

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

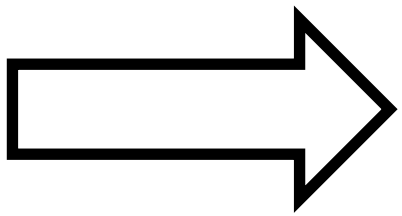
(credits: Thorsten Berger’s slide)

NOKIA
Connecting People

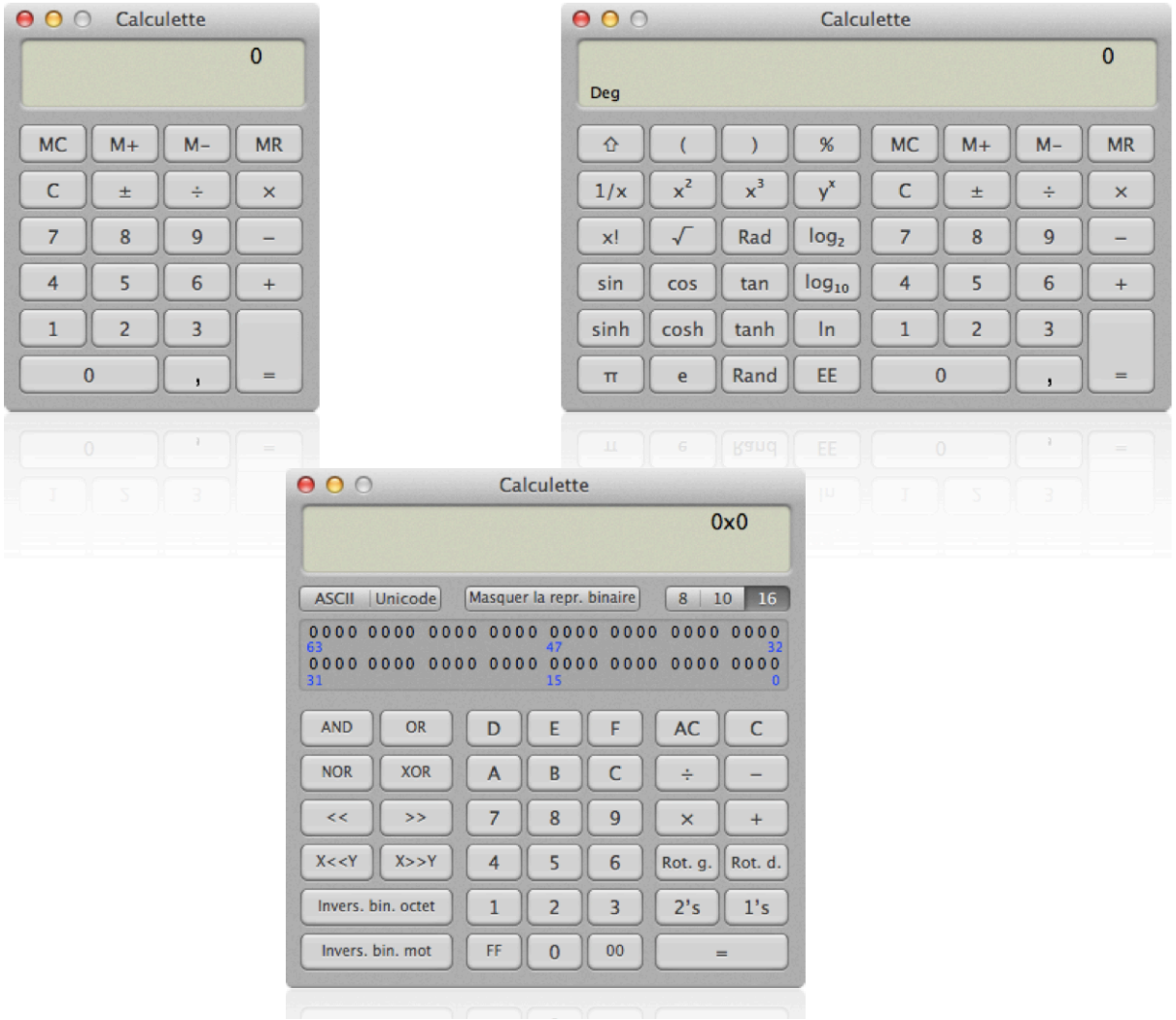
Know our past. Create the future...



Software-intensive systems



come in many **variants**


Calculette
0
MC M+ M− MR
C ± ÷ ×
7 8 9 −
4 5 6 +
1 2 3 =
0 ,
Calculette
0
Deg
⇧ ( ) % MC M+ M− MR
1/x x² x³ yˣ C ± ÷ ×
x! √ Rad log₂ 7 8 9 −
sin cos tan log₁₀ 4 5 6 +
sinh cosh tanh ln 1 2 3 =
π e Rand EE 0 ,
Calculette
0x0
ASCII Unicode
Masquer la repr. binaire
8 10 16
0000 0000 0000 0000 0000 0000 0000 0000
63 47 32
0000 0000 0000 0000 0000 0000 0000 0000
31 15 0
AND OR D E F AC C
NOR XOR A B C ÷ −
<< >> 7 8 9 × +
X<<Y X>>Y 4 5 6 Rot. g. Rot. d.
Invers. bin. octet 1 2 3 2's 1's
Invers. bin. mot FF 0 00 =

« 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)

Developer Tools
Development
Drivers
DTP/Prepress
Educational
Finance
Font Tools
Games
Graphics
HTML Tools
Internet Utilities
IPhone Applications
IPod Tools
Math/Scientific
Multimedia
Network/Admin
Screensave
Security
Spot t Plug Utilities
System ilities
Video
Word Processing

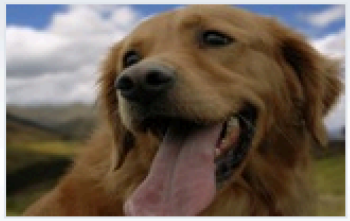
LOBAL PAGES >>
EWS ARCHIVE >>
OFTPEDIA REVIEWS >>
EET THE EDITORS >>

Power Matte 2 1.3 update

Adobe Ar plugin that can extract a ect in an i e

[read

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



Grid 1.1 update

Helps you generate perspective grids

[read more >]

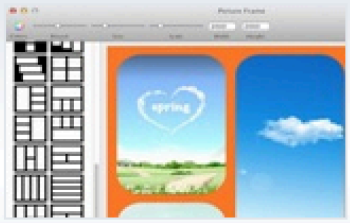
Size:	102 KB
Platform:	Mac OS X 10.8 or later
License:	Commercialware
Rating:	NOT RATED
Downloads:	21
Updated:	June 20th, 07:56 UTC

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 l...
License:	Commercialware
Rating:	Excellent (5.0/5)
Downloads:	297
Updated:	June 20th, 07:53 UTC

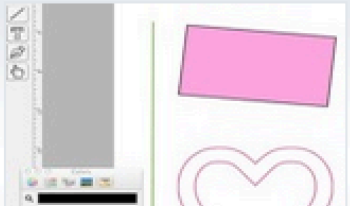


FashionLab Studio 1.1 update

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 l...
License:	Commercialware
Rating:	NOT RATED
Downloads:	3
Updated:	June 20th, 07:49 UTC



Variability

RENAULT

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

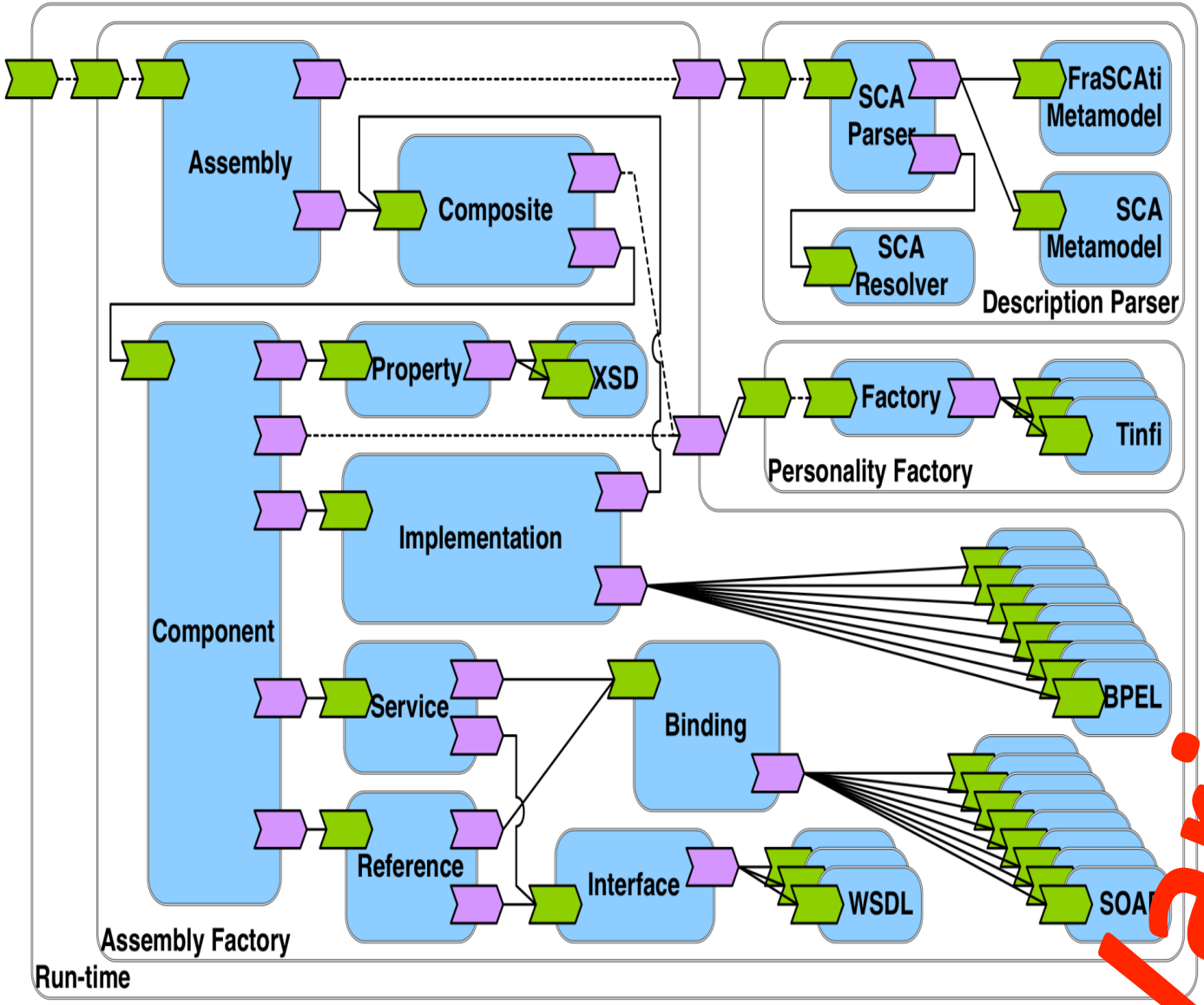
01 Preferences	02 Version	03 Equipment & options

Previous Next

OPTIONS

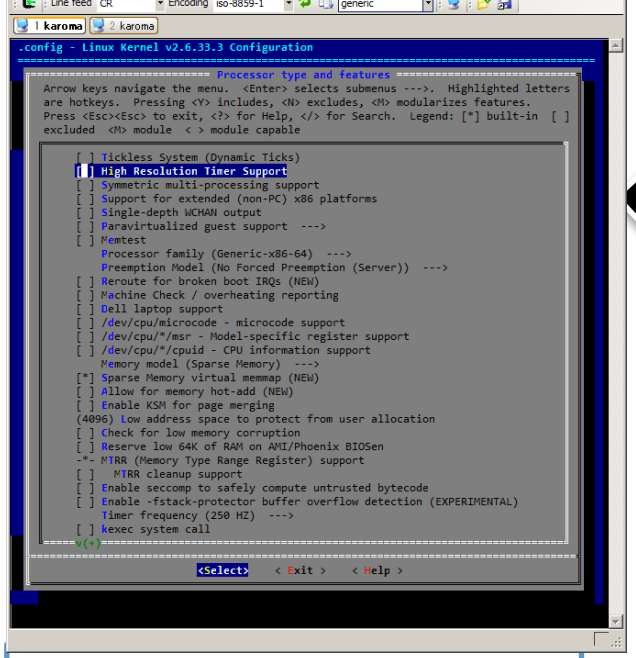
> COMFORT	
☑ Central storage console & armrest between seats	£50.00
> DRIVING	
☐ Electric door mirrors	£0.00
> SAFETY & SECURITY	
☑ ESC (Electronic Stability Control) with traction and understeer control	£200.00

Variability

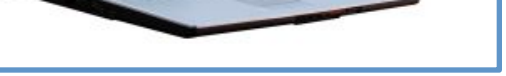


Variability

Printer Firmware



Linux Kernel



Features C

Service name A	Automatic forwarding	E-mail client access[14]	client E-mail for other server	Integration with IM service B	Domain Name customization	Interface script technique
AOL Mail	No	Yes (POP3, IMAP, SMTP)	Yes[0]	AOL Instant Messenger	No[1]	JavaScript/ Ajax 4
Bigfoot Communications	Premium account only	Yes (POP3, IMAP, SMTP)	Yes (POP3 only)	XMPP 3	Yes	HTML/ JavaScript/ CSS/Ajax
FastMail.FM	Paid accounts only	Yes (IMAP)[7] 2	Paid accounts (POP3, Hotmail)	XMPP	Enhanced and group (Business/ Family) accounts	HTML/ JavaScript/ CSS/Ajax (Optional user supplied custom css+JavaScript)
Gmail	Yes	Yes (POP3, IMAP) SSL/TLS supported SMTP restricted[18]	Yes (POP3 only)	Google Talk[beta] (XMPP), AOL Instant Messenger	Yes (Google Apps $5.00 monthly/ $50.00 annually)	HTML/ JavaScript/ Ajax[2]
GMX Mail	No	Yes (POP3, IMAP[17], SMTP) SSL/TLS supported	Yes (POP3 only)	XMPP	Yes	HTML/ JavaScript/ Ajax
Hushmail	No	Extra cost[8]	? 5	No	$1.99/$3.99 monthly through Hushmail Business	Java or HTML
Mail.com	No	Yes (POP3, IMAP, SMTP) SSL/TLS supported	Yes (POP3 only)	Google Talk (XMPP)	No	HTML/ JavaScript/ Ajax[2]
Mail.ru	Yes	Yes (POP3, IMAP)	Yes (POP3 only)	custom 7	?	HTML/ Ajax (Beta)
rediff	No	Plus members only	?	Rediff Bol	Yes 1	JavaScript/ Ajax[2]
Runbox	Yes	Yes (IMAP, POP, SMTP) SSL/TLS supported	Yes (POP3, Hotmail, Gmail) SSL/TLS supported	XMPP, Google Talk, AOL Instant Messenger, MSN, ICQ, IRC[41]	Yes	HTML/ JavaScript/ CSS/Ajax
Seznam.cz	Yes	Yes (POP3, IMAP, SMTP) SSL/TLS supported	Yes (POP3 only)	No	No	HTML/ JavaScript
Windows Live Hotmail	Yes	Partial (POP3, SMTP)[3]	Yes (POP3 only)	Windows Live Messenger	Yes[4]	HTML/ JavaScript/ CSS/Ajax
Yahoo! Mail	Plus accounts only	Yes (POP3-Plus members only, but free in some countries, IMAP SSL/TLS supported)	6	Yes (POP3 only)	Yahoo! Messenger, Windows Live Messenger	$35 yearly 7
Yandex Mail	Yes	Yes (POP3, IMAP, SMTP, SSL)	Yes (POP3 only)	Ya Online, any XMPP IM	Yes (Free, Yandex PDD supports up to 1000 mailboxes without verification of legal use)	HTML/ JavaScript/ CSS/Ajax

Variability Patterns

1. Boolean yes/no answers
2. Partial/constrained yes/no answers
3. Single-value answers
4. Multiple values answers
5. “Unknown” answers
6. Empty cells
7. Inconsistent cells
8. Additional / Extra information

Variability

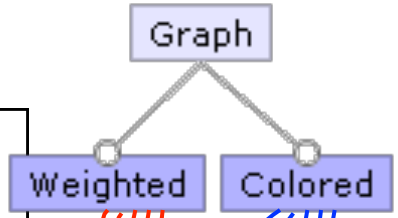
```
class Graph {
  Vector nv = new Vector(); Vector ev = new Vector();
  Edge add(Node n, Node m) {
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); ev.add(e);
    e.weight = new Weight();
    return e;
  }
  Edge add(Node n, Node m, Weight w)
    Edge e = new Edge(n, m);
    nv.add(n); nv.add(m); 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();
  }
}
```

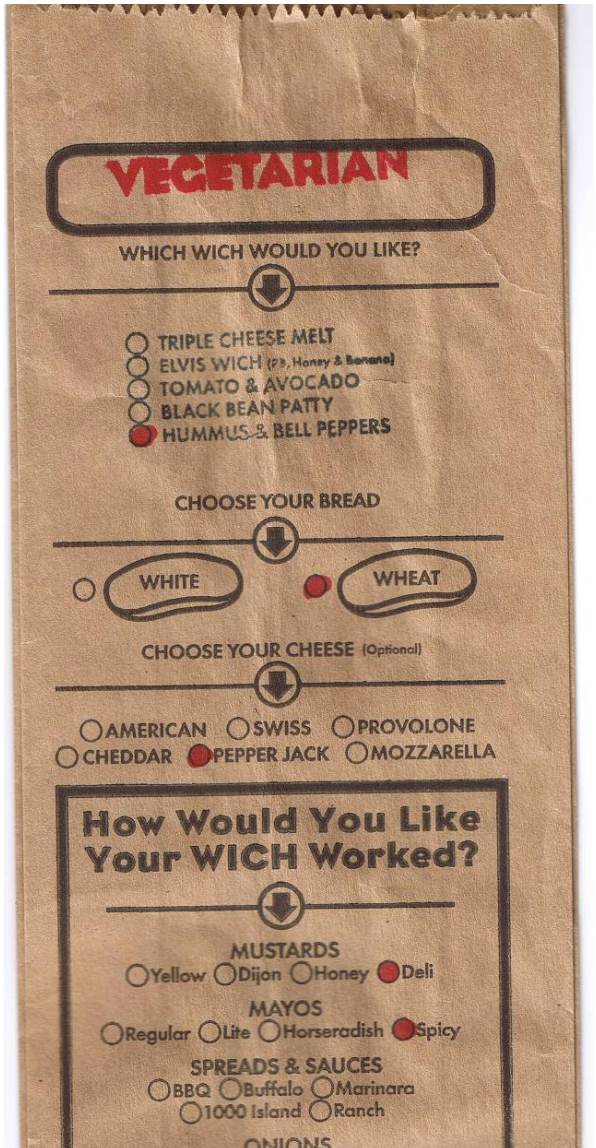
```
class Color {
  static void setDisplayColor(Color c) { ... }
}
```

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



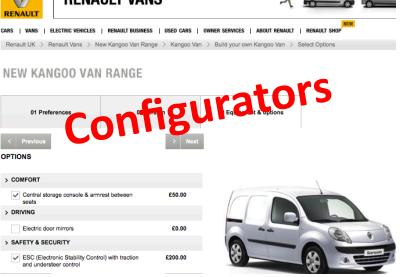
(credits: Christian Kaestner's slide)

Food? Product lines!

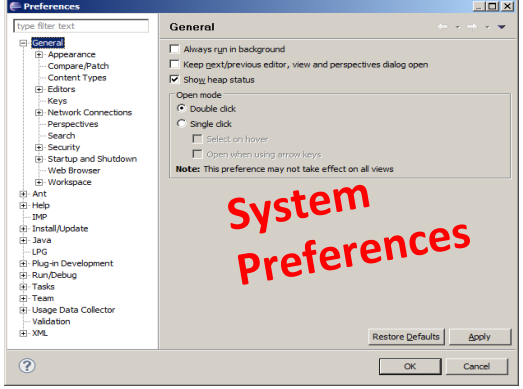



VEGETARIAN
How Would You Like Your WICH Worked?
VEGETARIAN
How Would You Like Your WICH Worked?

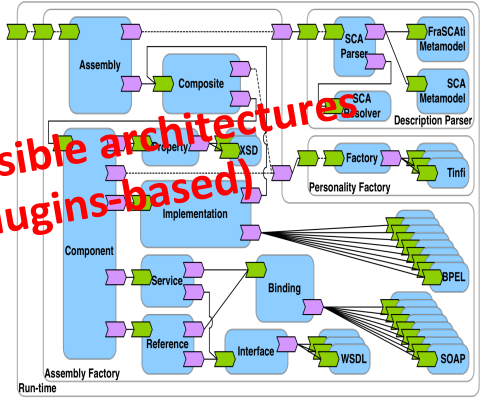
Product Lines and Variability



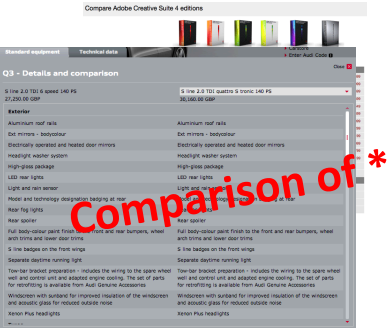
Configurators



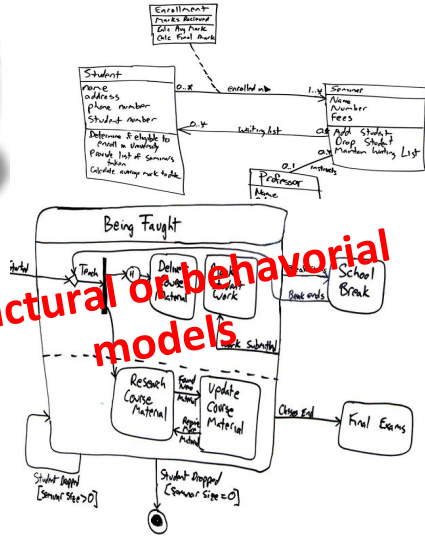
System Preferences



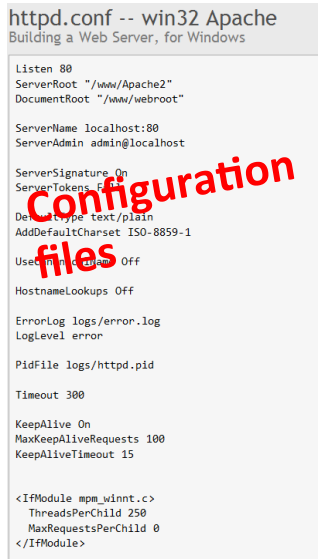
Extensible architectures (eg plugins-based)



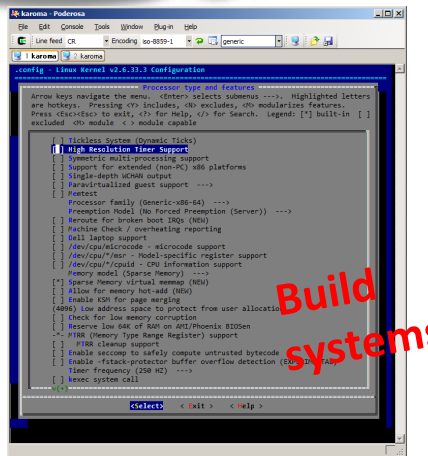
Comparison of *



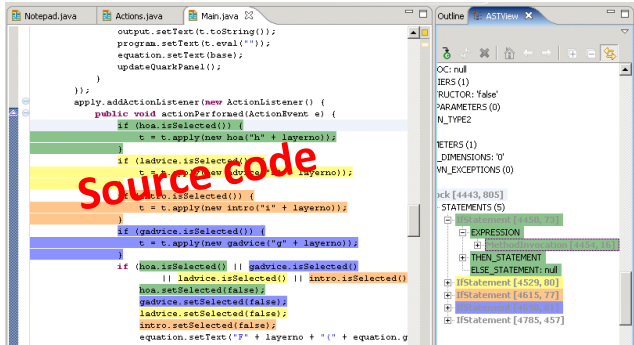
Structural or behavorial models



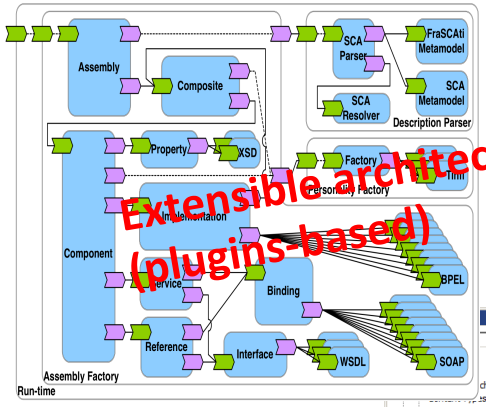
Configuration files



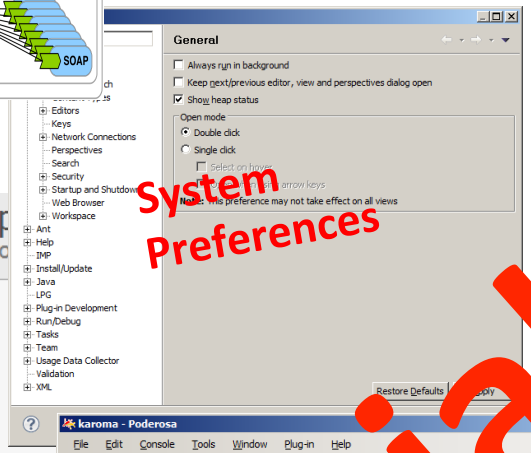
Build systems



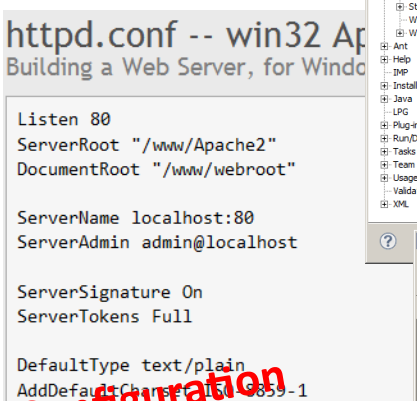
Source code



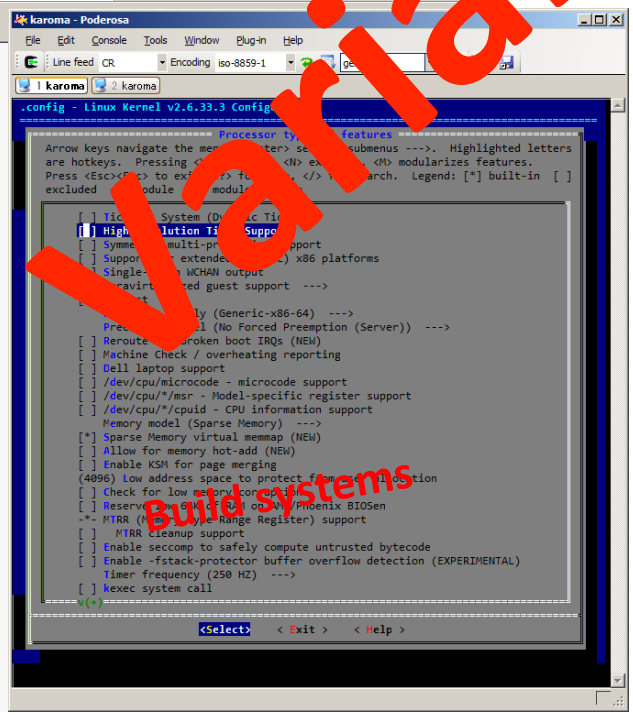
Extensible architectures (plugins-based)



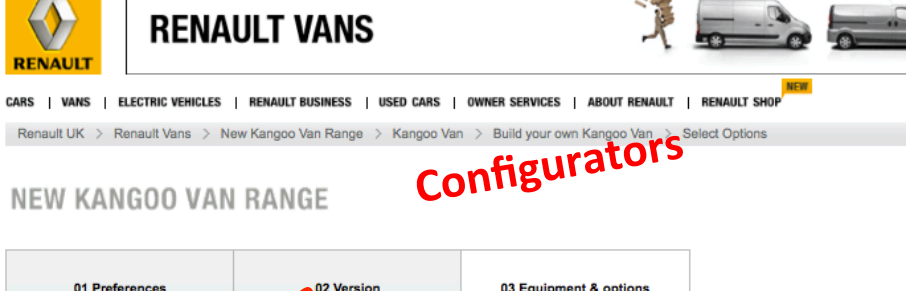
System Preferences



Configuration files



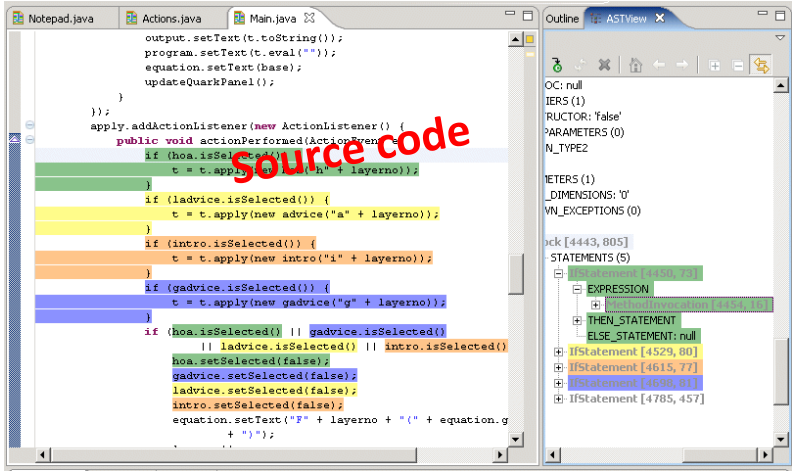
Build systems



Configurators

Variability

Comparison of Product



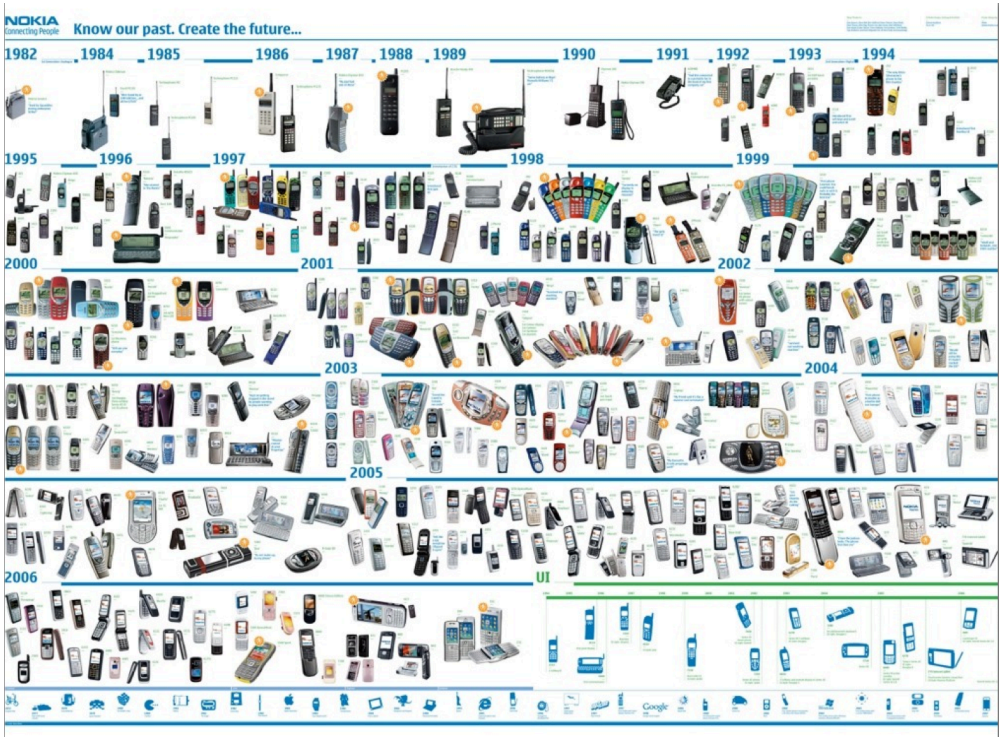
Source code

Variability Engineering

Challenges, Promises, and Benefits

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

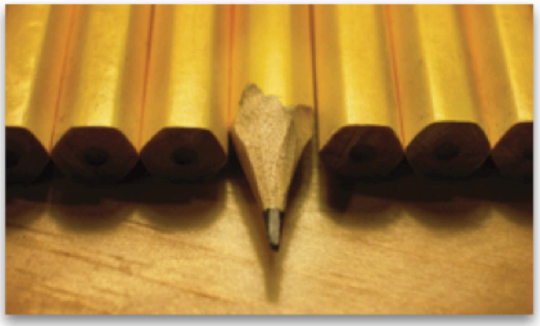


Benefits

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

BOSCH
Invented for life

PHILIPS

ERICSSON

NOKIA
Connecting People

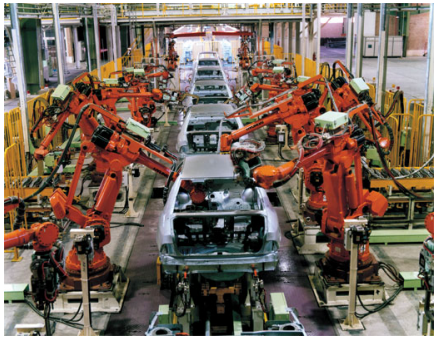
CelsiusTech

Lucent Technologies
Bell Labs Innovations

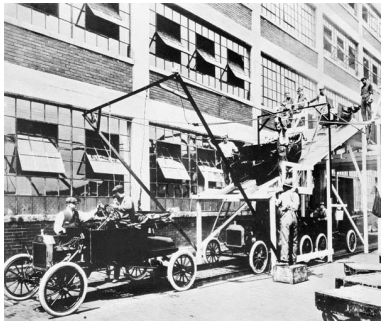
Printer Firmware

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

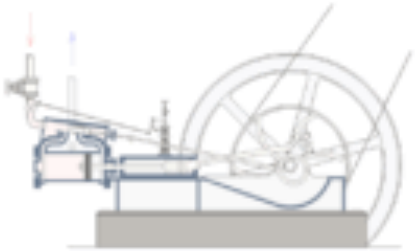
A Bit of History: Industrial Revolution



1980s



1901
Henry Ford



1698
Thomas Savery

Nowaday: Product Lines Everywhere

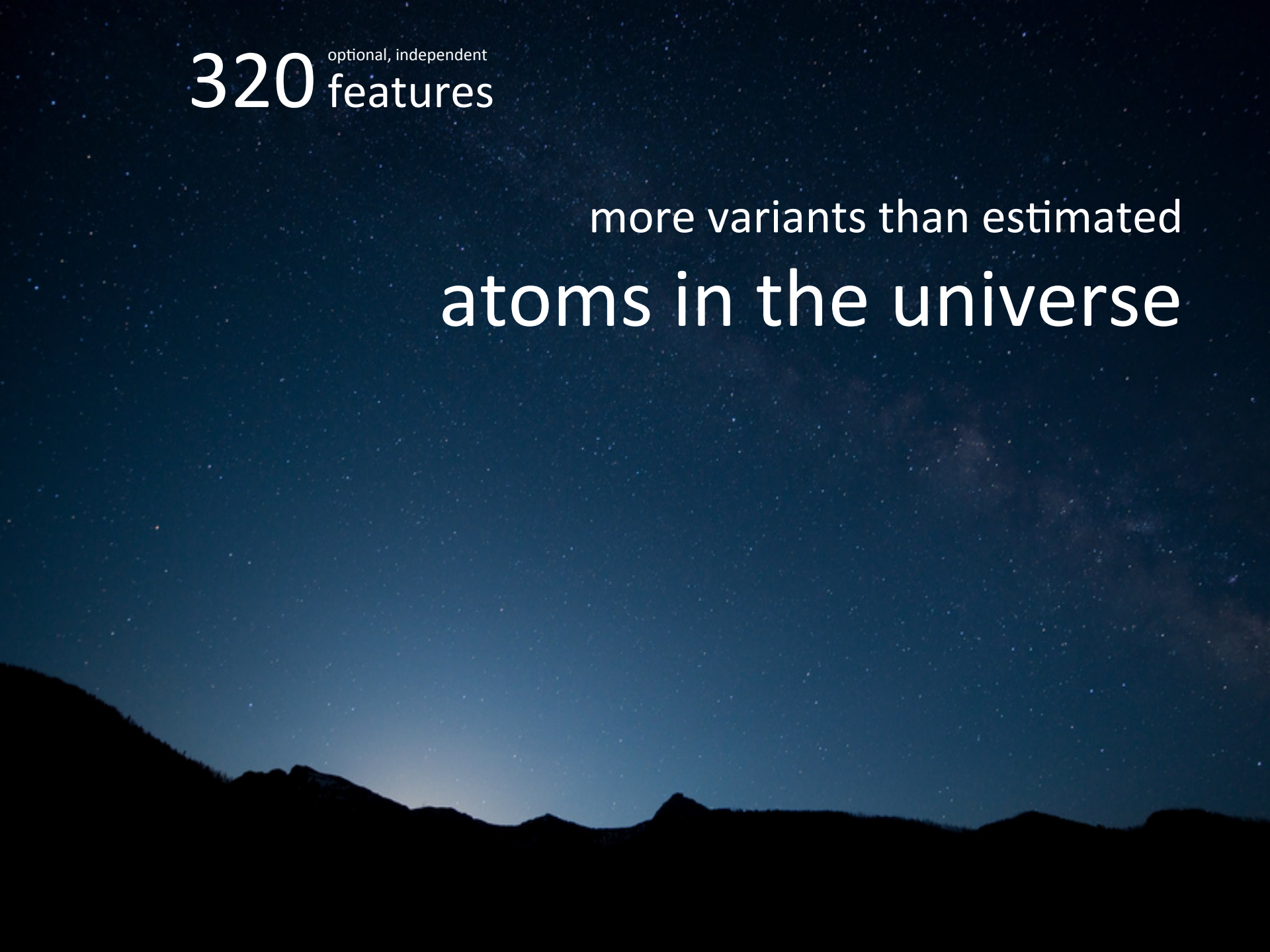

Variability = Complexity

(credits: Christian Kaestner's slide)

33 optional, independent features

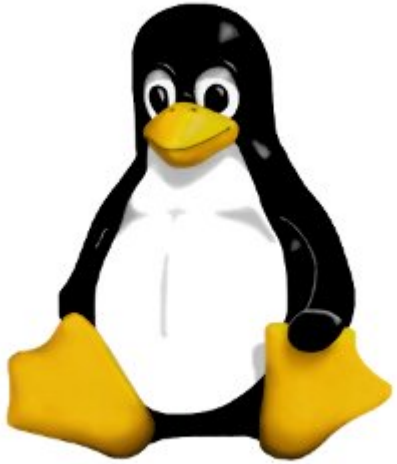
a unique variant for every

person on this planet


320 optional, independent features
more variants than estimated
atoms in the universe

2000 features

10000 features



Software Product Line and Variability Engineering

A revisit of your cursus

What is new?

Family vs single systems

Focus on reuse

Domain engineering

Factoring out commonality

Managing variability

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.intreqr, 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);
.......
```

cyberstormII.c

```
....
static void dma_dump_state(struct NCR_ESP *esp)
{
        ESPLOG(("esp%d: dma -- cond_reg<%02x>\n",
                esp->esp_id, ((struct cyberII_dma_registers *)
                                        (esp->dregs))->cond_reg));
        ESPLOG(("intreq:<%04x>, intena:<%04x>\n",
                custom.intreqr, custom.intenar));
}

static void dma_init_read(struct NCR_ESP *esp, __u32 addr, int
        length)
{
        struct cyberII_dma_registers *dregs =
                (struct cyberII_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

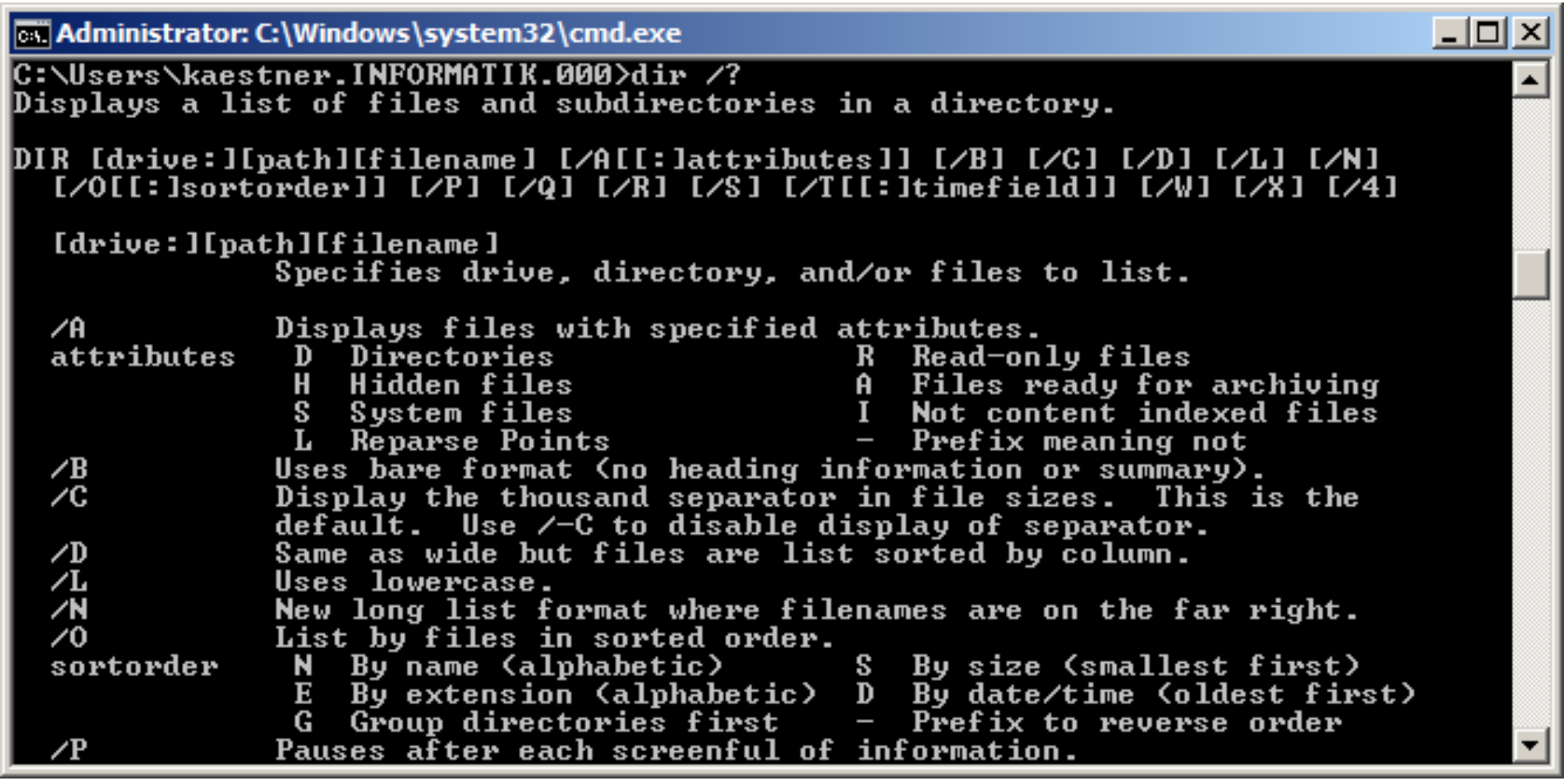
- Hard to propagate bug fixes

~ Ensure consistent behavior of all clones

« variability »

Is it really new?

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[[:]attributes]] [/B] [/C] [/D] [/L] [/N]
[/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]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_c
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     }
```

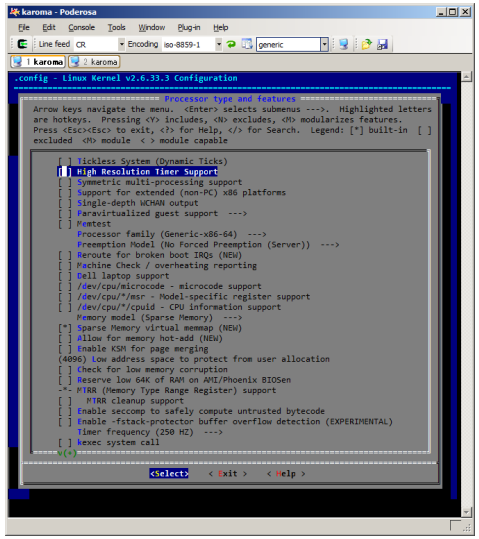
Global configuration

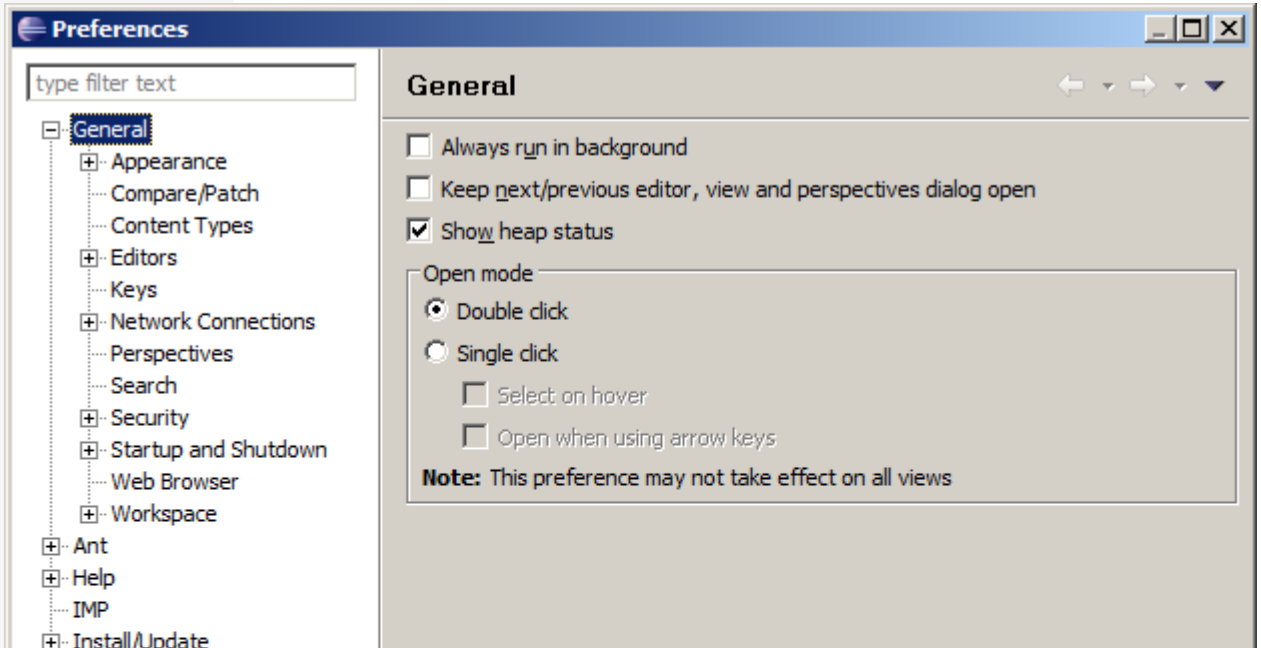
```
class Config {
  public static boolean isLogging = false;
  public static boolean isWindows = false;
  public static boolean isLinux = true;
}
class Main {
  public void foo() {
    if (isLogging)
      log(„running foo()“);
    if (isWindows)
      callWindowsMethod();
    else if (isLinux)
   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

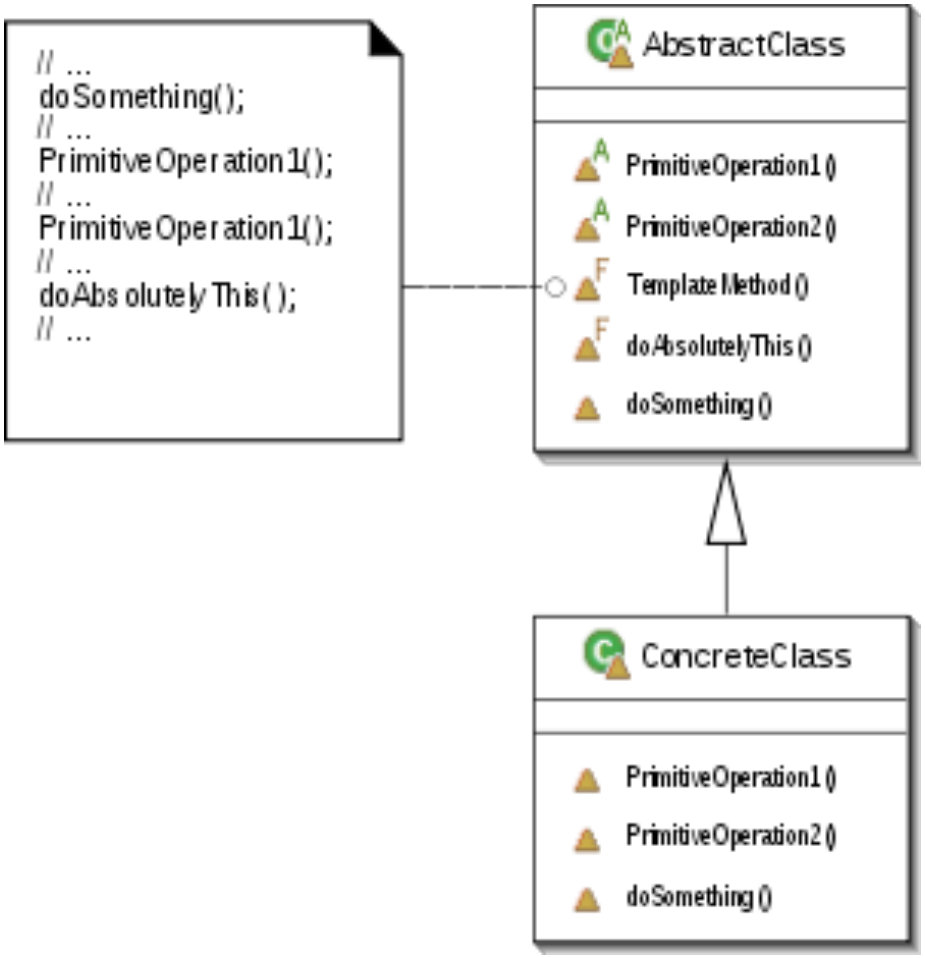
Factory

Strategy

Decorator

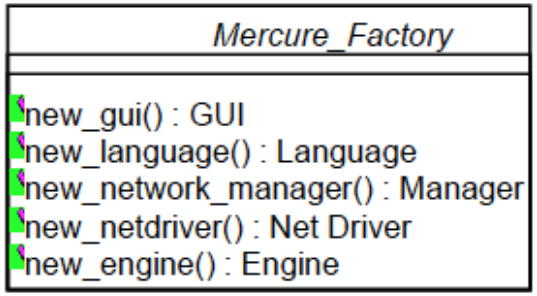
....

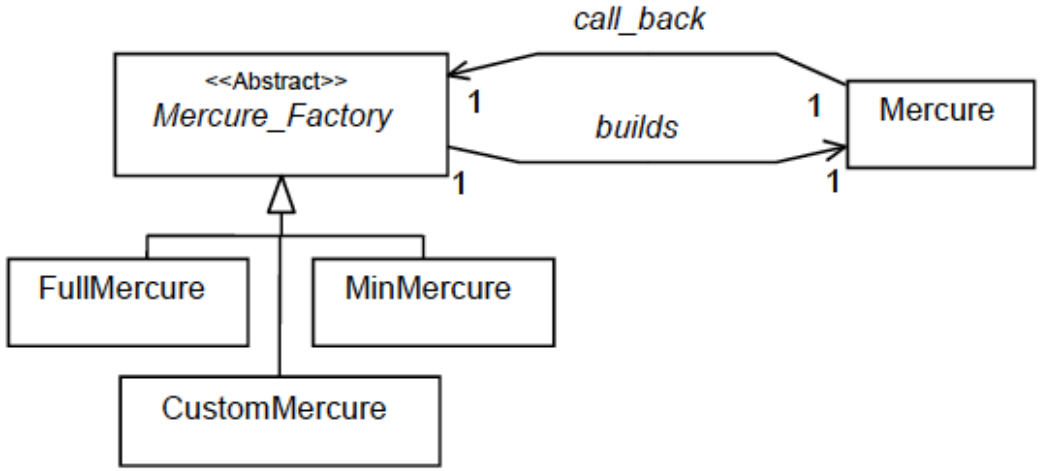
Template Method

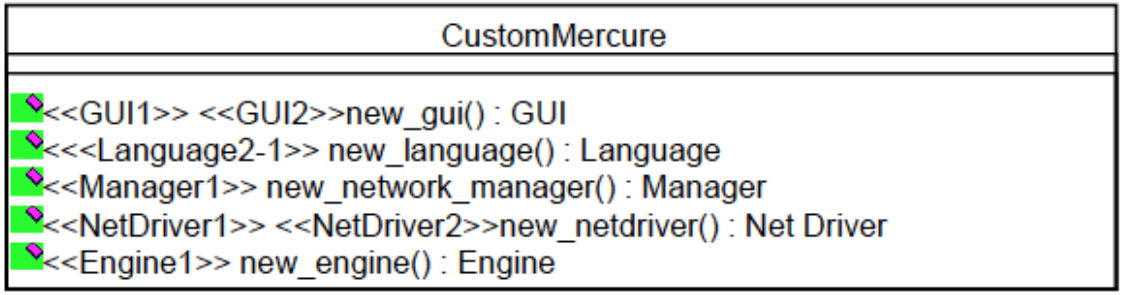


The decision model

- The Abstract Factory Design Pattern
  - [Gamma et al 95]







API

Framework

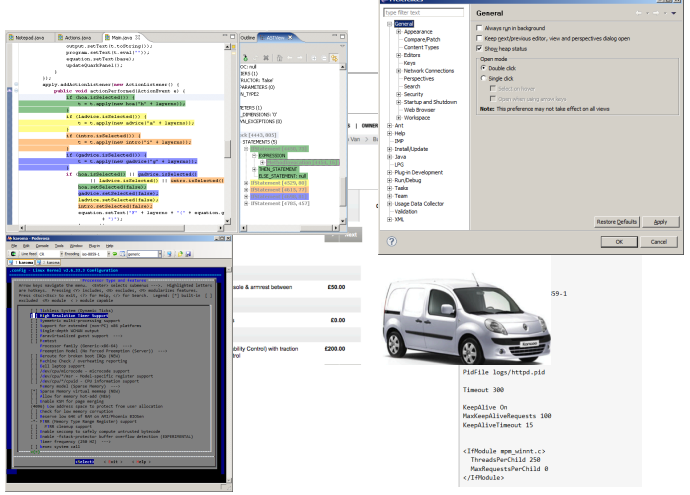
Plugin-based systems

(Active) Annotations

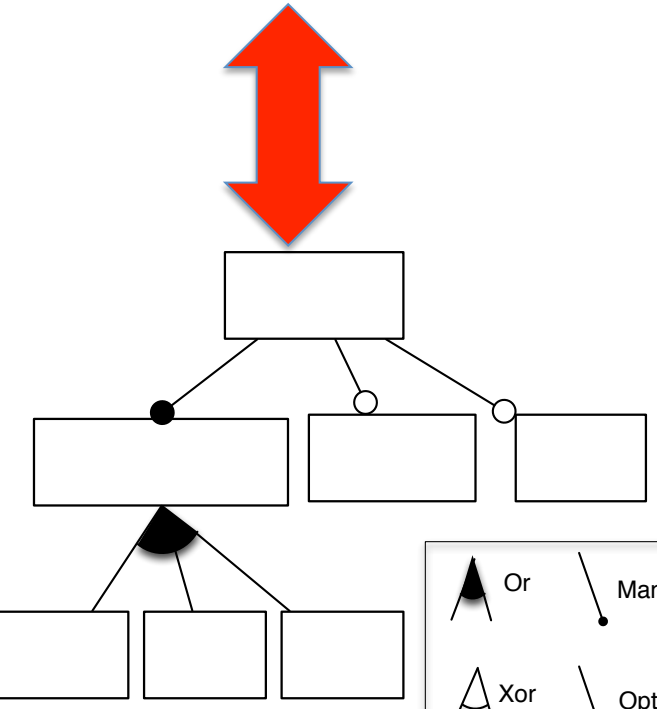
can have parameters

Metamodeling and Domain-Specific Languages

Variability (general approach)



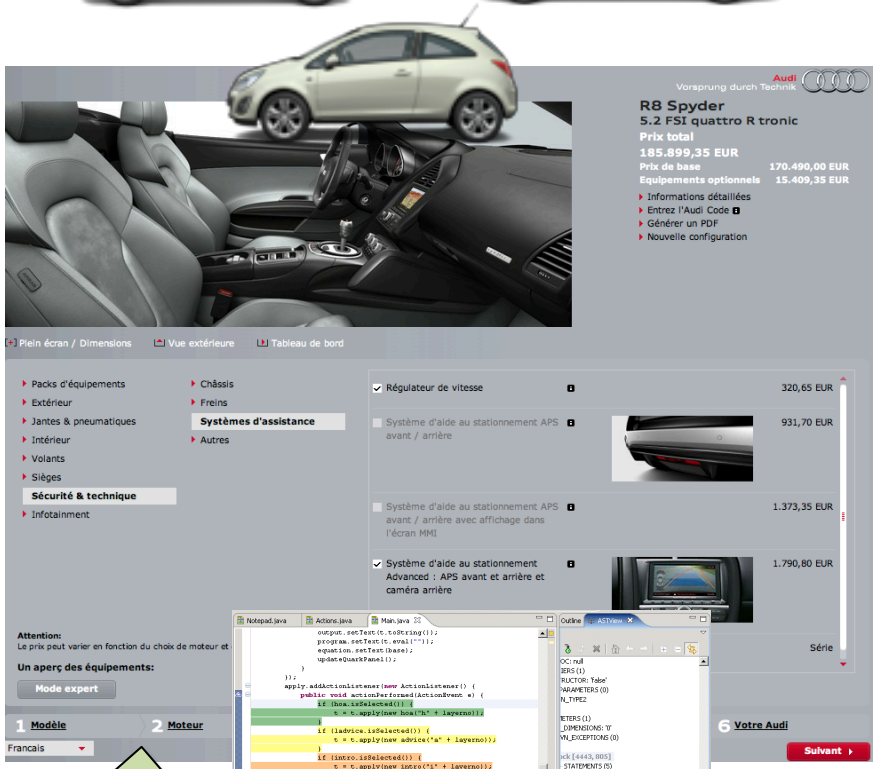
Variants of code (e.g., Java ou C)
Variants of user interfaces
Variants of video sequences
Variants of models (e.g., UML or SysML)
...

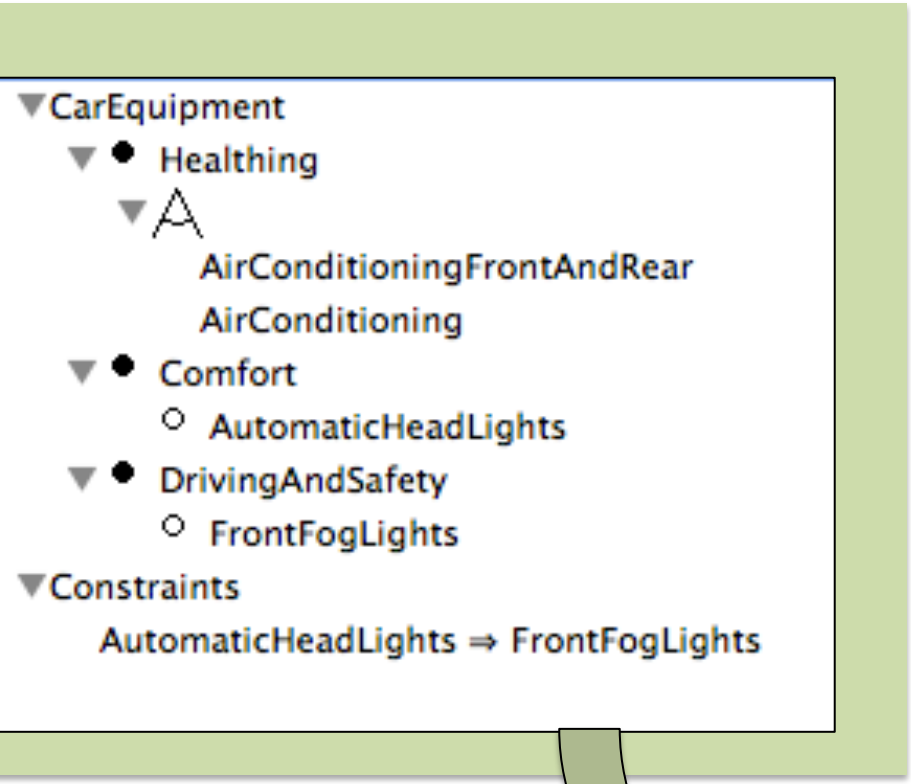


not, and, or, implies

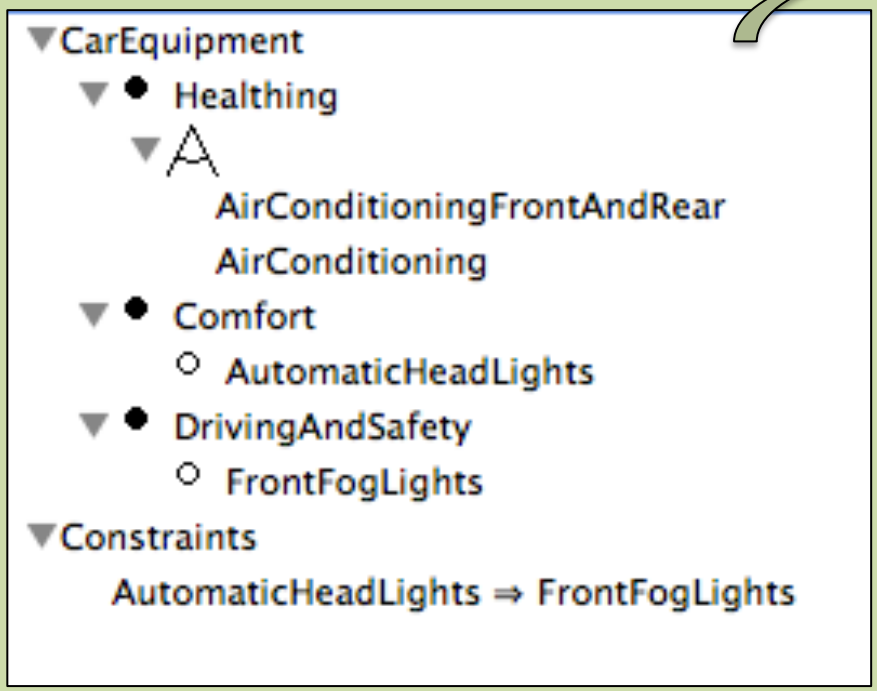
Variability Models (feature models)

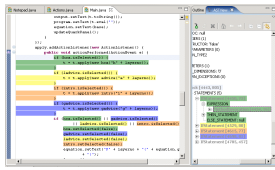
Feature Models





Feature Models (Background)

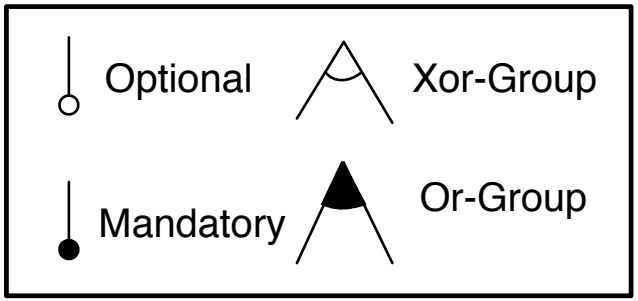




Hierarchy: rooted tree

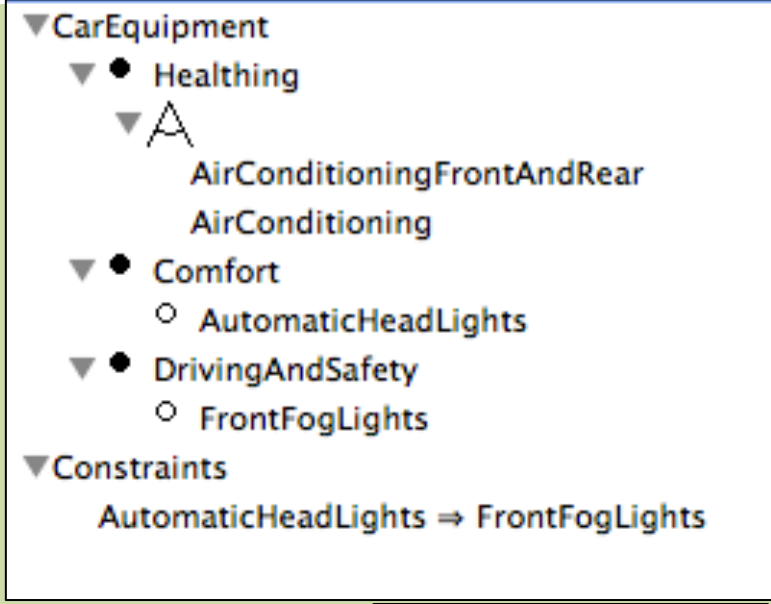
Variability:

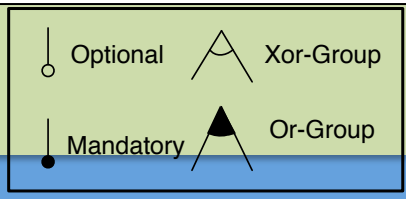
- mandatory,
- optional,
- Groups: exclusive or inclusive features
- Cross-tree constraints



SPLC12Scripts Model | SPLC12Scripts.config

- CarEquipment (valid, 1 possible configurations)
  - Healthing
    - AirConditioningFrontAndRear
    - AirConditioning
  - Comfort
    - AutomaticHeadLights
  - DrivingAndSafety
    - FrontFogLights





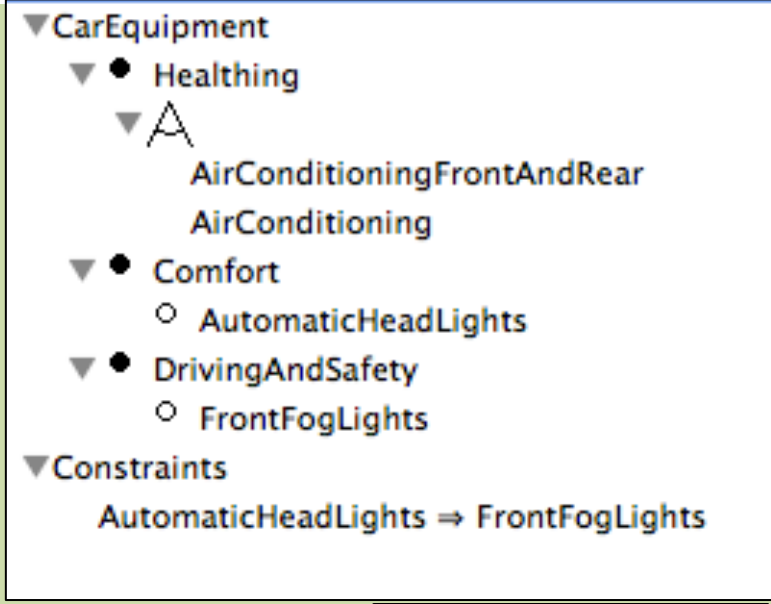
Hierarchy + Variability
=
set of valid configurations

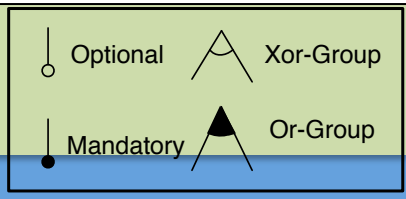
configuration = set of features selected

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

SPLC12Scripts Model | SPLC12Scripts.config

- CarEquipment (valid, 1 possible configurations)
  - Healthing
    - AirConditioningFrontAndRear
    - AirConditioning
  - Comfort
    - AutomaticHeadLights
  - DrivingAndSafety
    - FrontFogLights

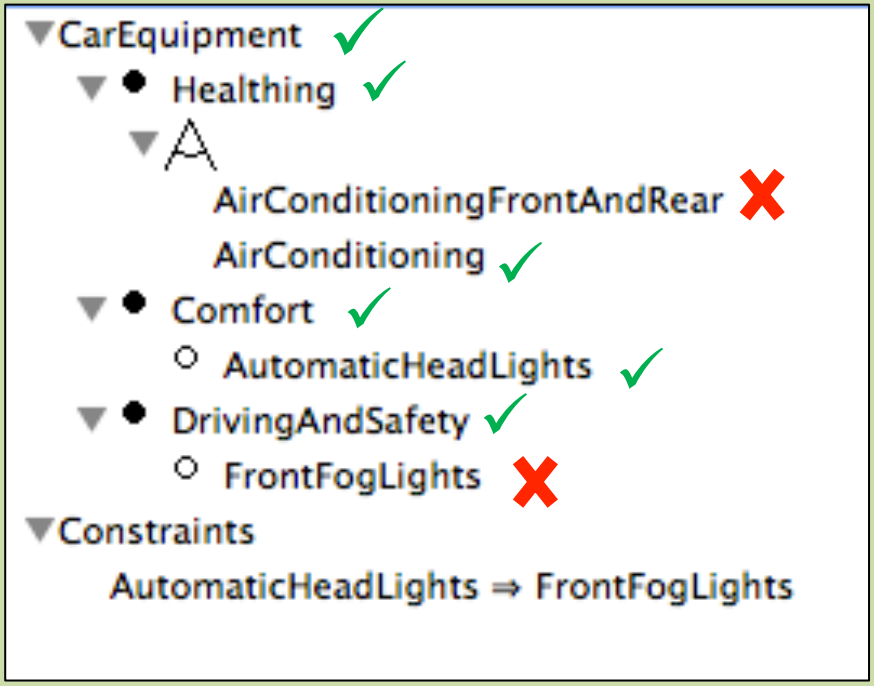


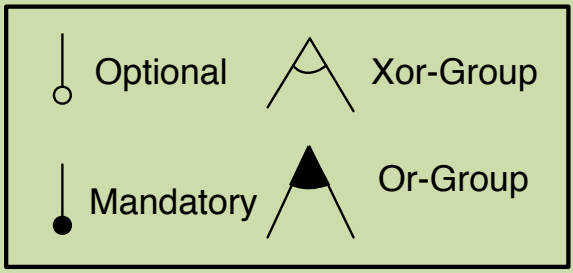


Hierarchy + Variability
=
set of valid configurations

configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning}

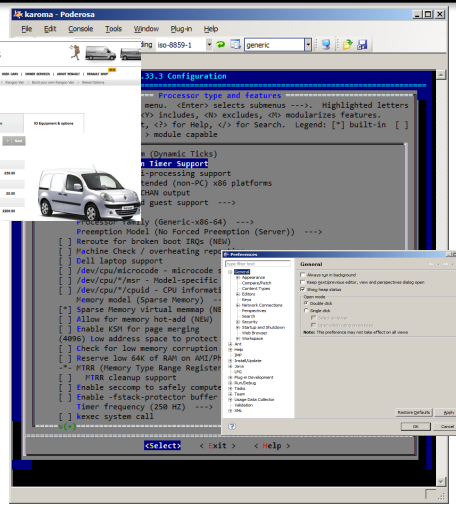




Hierarchy + Variability
=
set of valid configurations

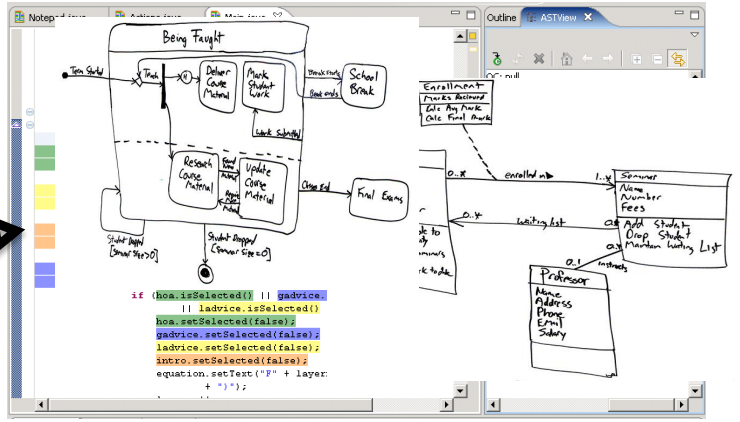
configuration = set of features selected

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



Feature Model

Variability Realization Model (VRM)



Base Artefacts (e.g., models)

Configuration

(resolution model)

Software Generator

(derivation engine)

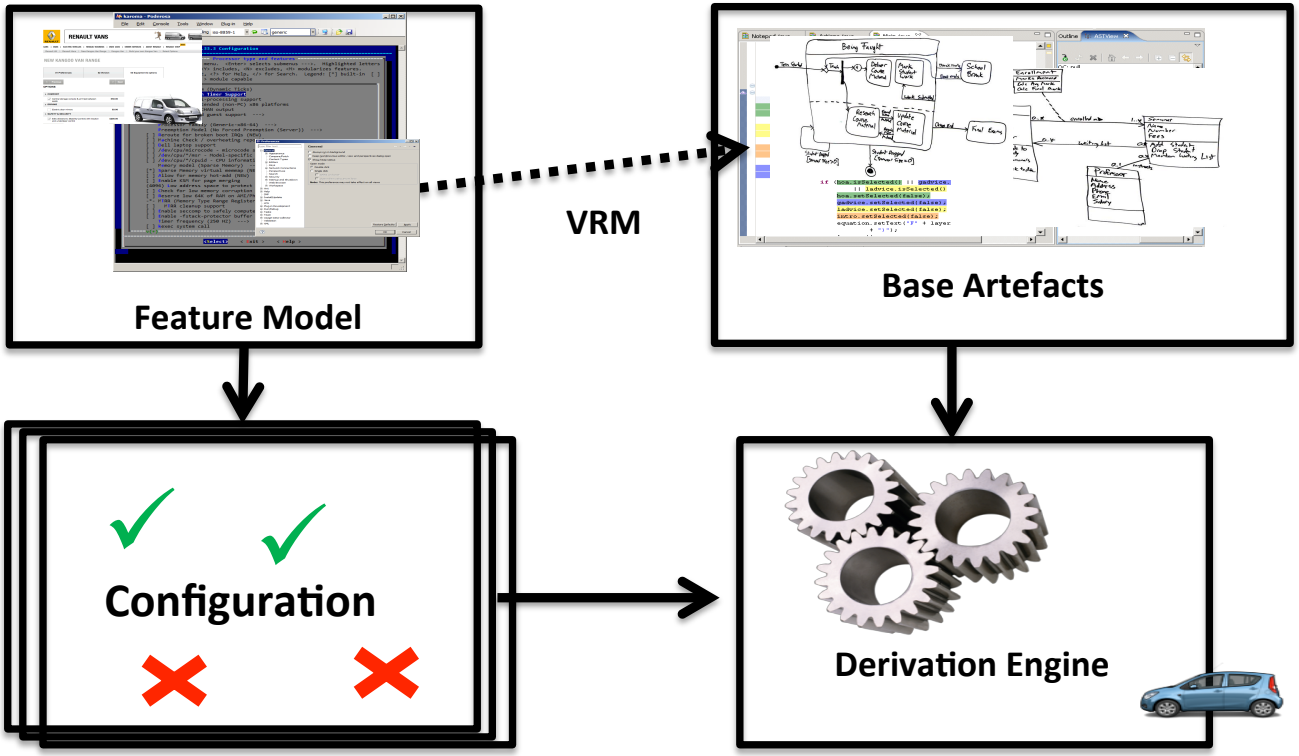

Unused flexibility

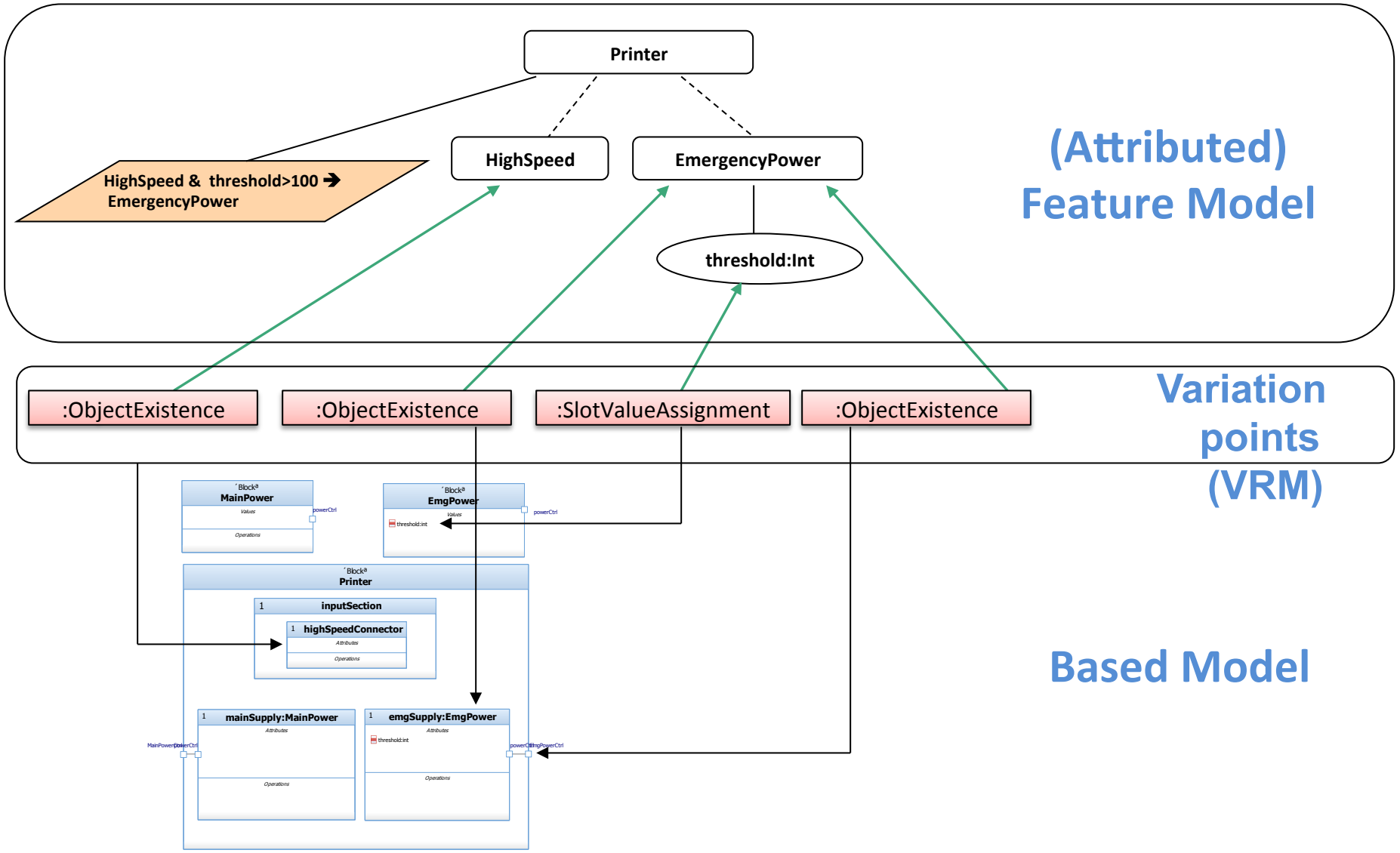

Illegal variant

Variability and Standardization

Three pilars of CVL

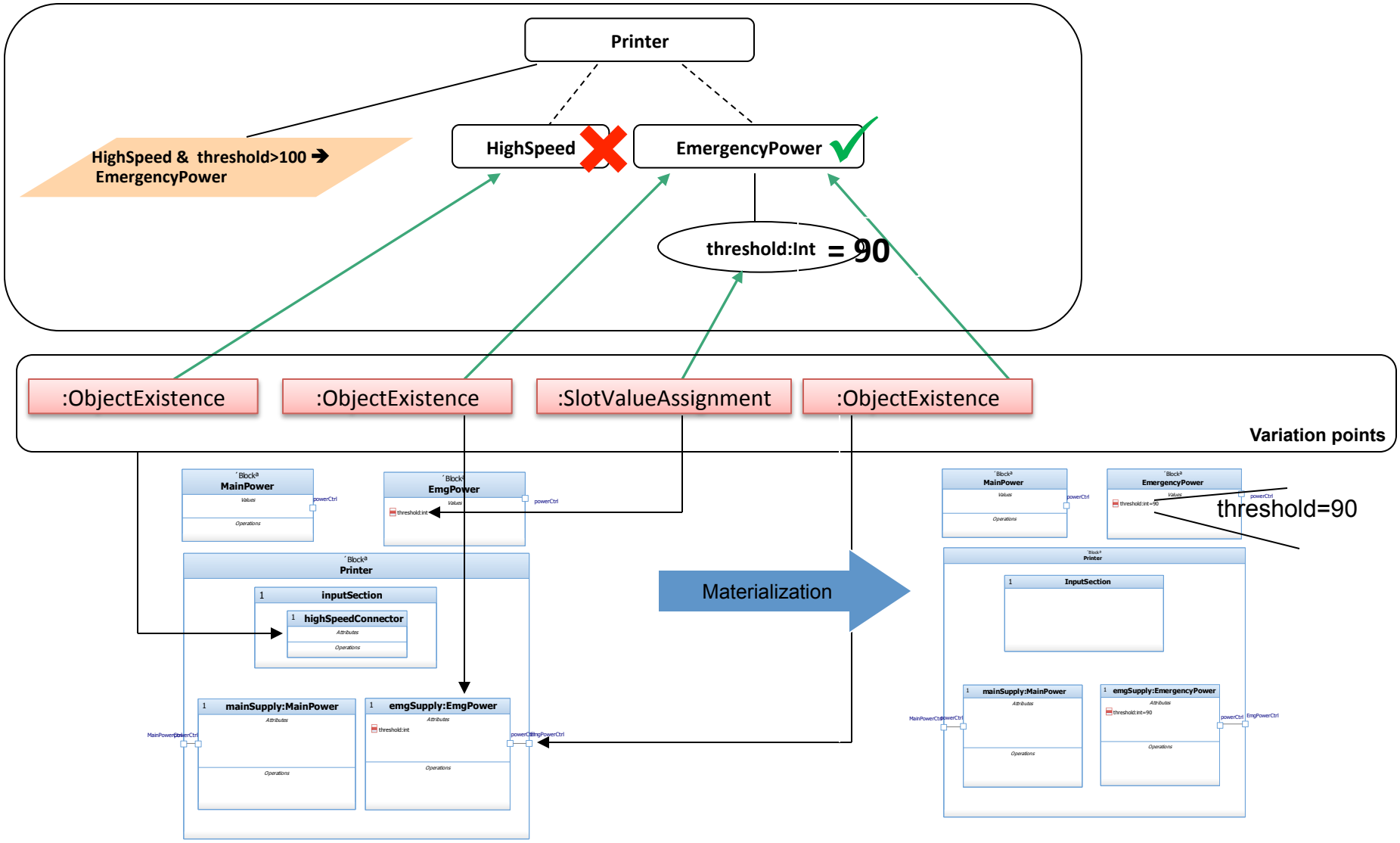
- **Feature Models** with attributes and multi-features
- **Base** ”model” (eg UML or SysML model)
- Variability **Realization** Model (VRM)
  - add, remove, substitute, parameterization





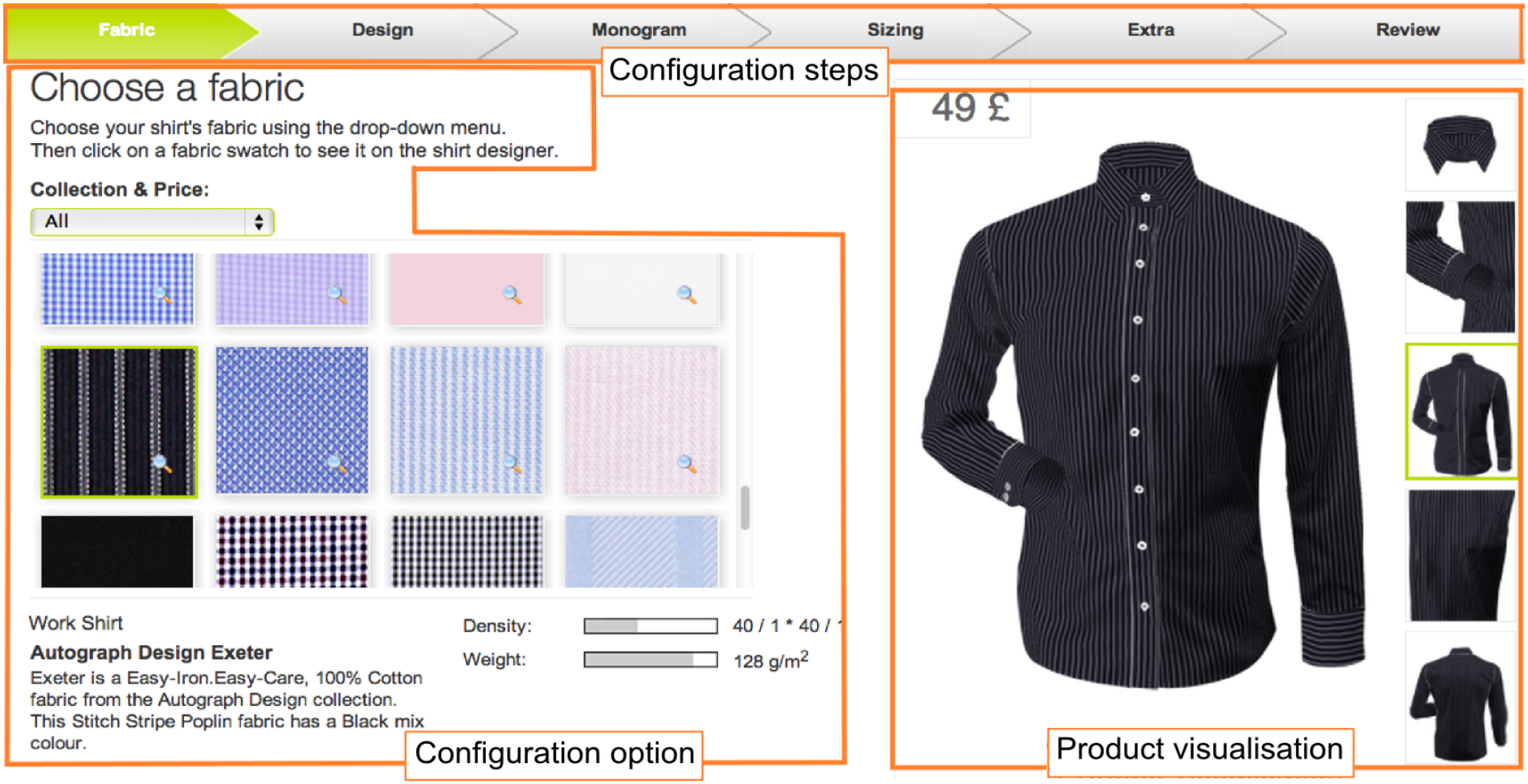
Adapted from the CVL tutorial at SPLC'12 by Oystein Haugen, Andrezj Wasowski, Krzysztof Czarnecki

Configuration and Product Derivation

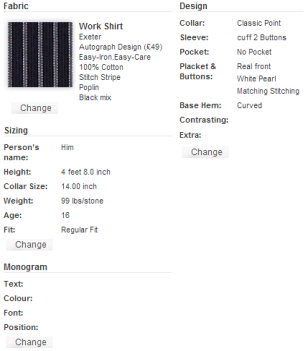


Web Configurators (visualization aspect)

How to render a product visualization given a set of options selected?

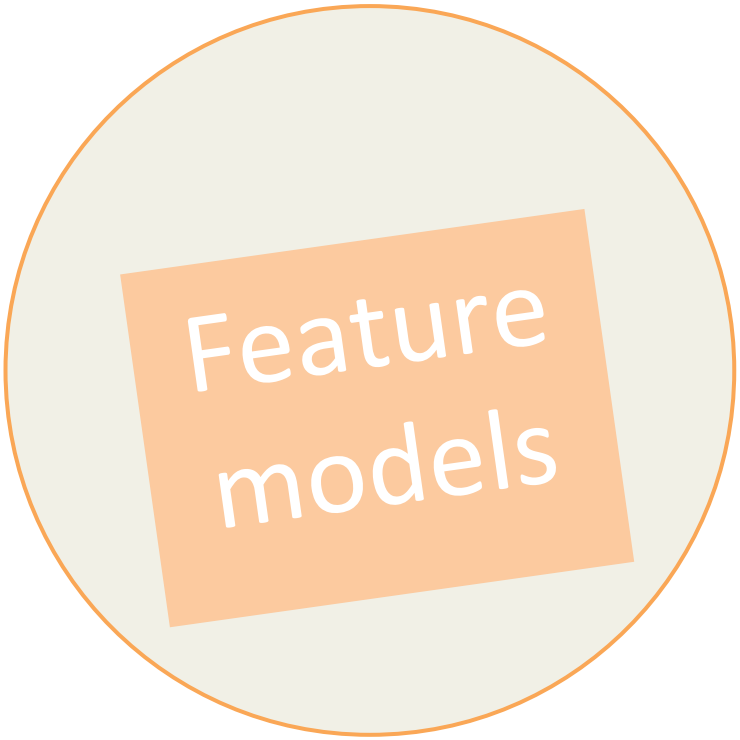


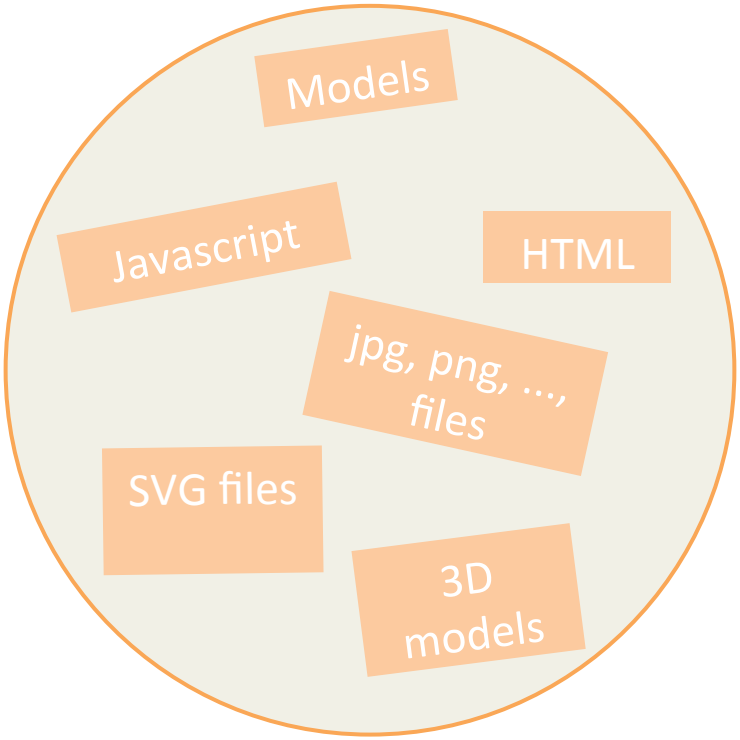
Marks & Spencer web configurator

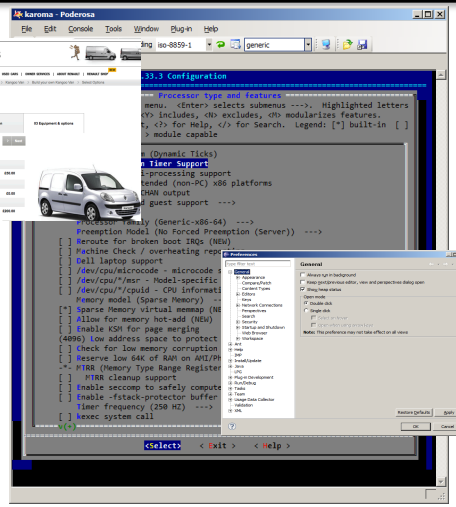


PRODUCT CONFIGURATION

VISUAL REPRESENTATION







Feature Model

Variability Realization Model (VRM)

Javascript

HTML

SVG files

Configuration

(resolution model)

Software

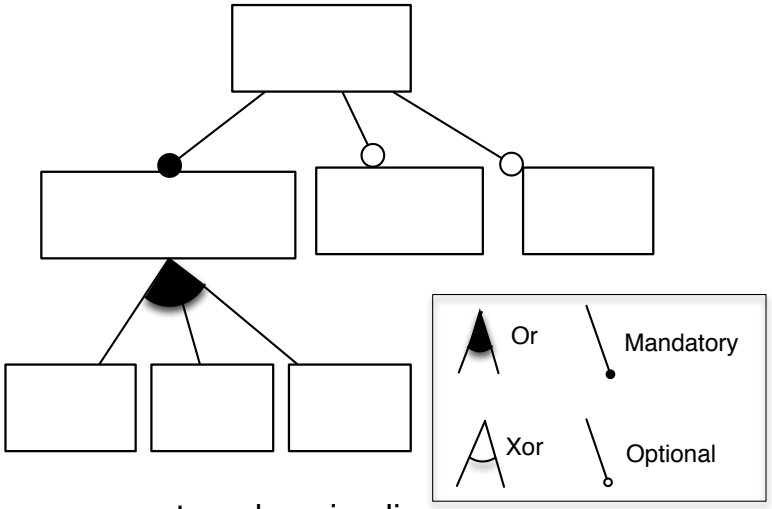
(derivati

Variability in the Video Domain

What are the differences?

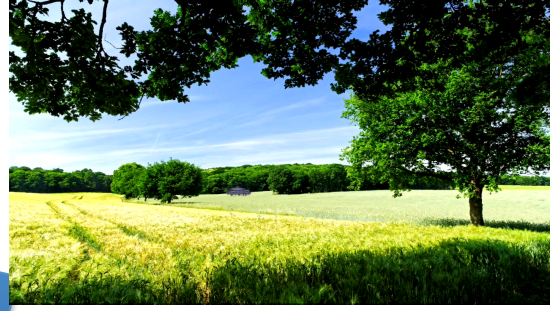
aka what is the variability of a video?

We synthesize video sequence variants with variability techniques



not, and, or, implies

Variability Models (feature models)



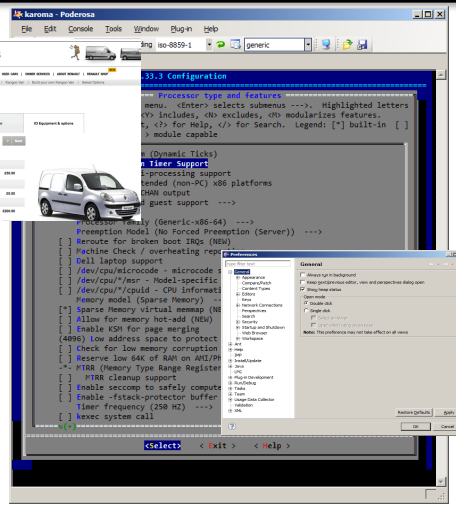
Why?

Industrial needs:
consumer and provider of video algorithms have severe difficulties to test their algorithms on different kinds of inputs.

Why?

Problem: collecting videos is a key economic problem.

Solution: hundreds of video sequences with different characteristics.



Feature Model

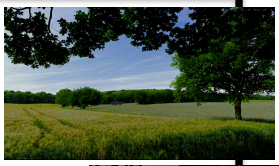
Variability Realization Model (VRM)

✓ ✓

Configuration

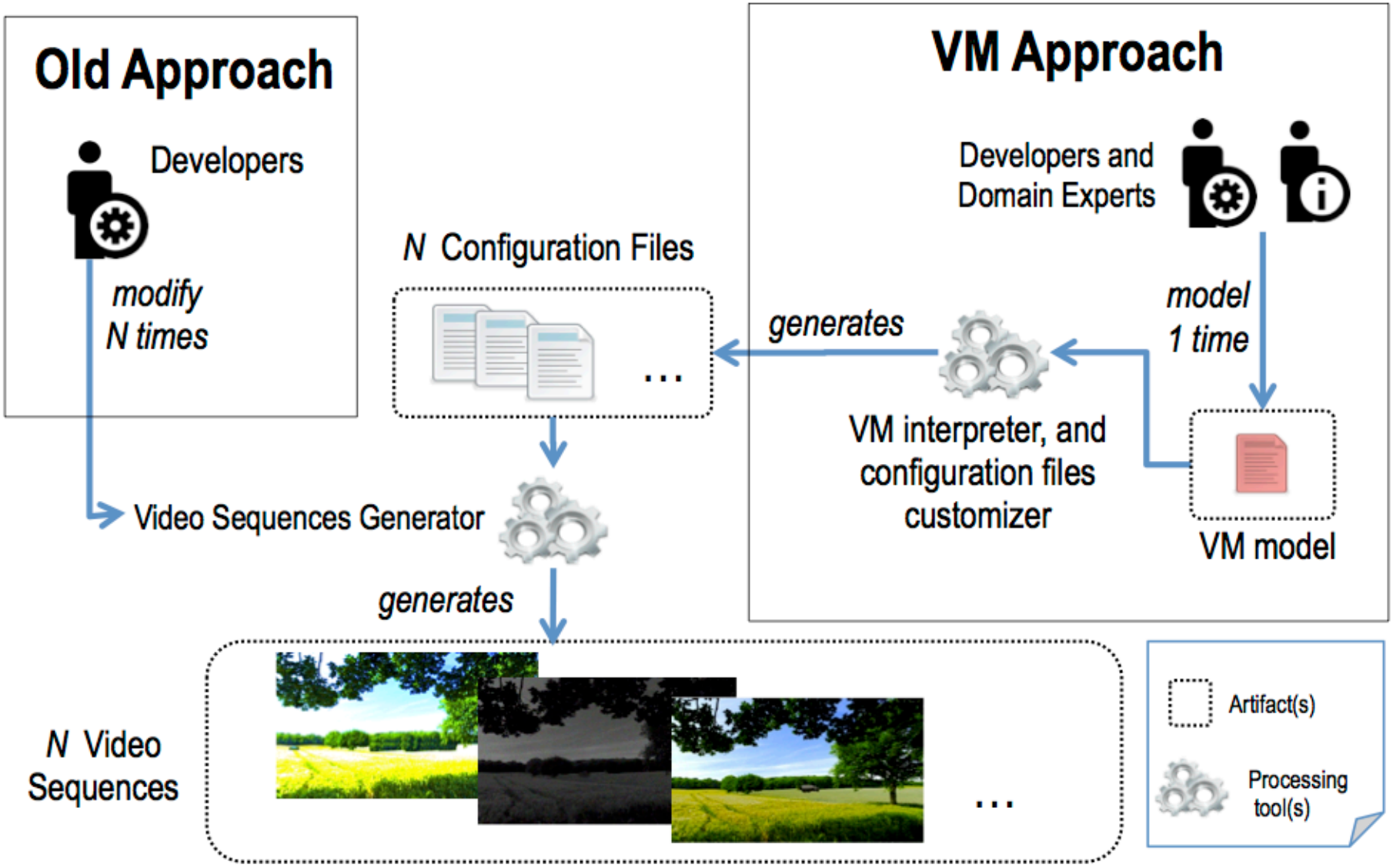
(resolution model)

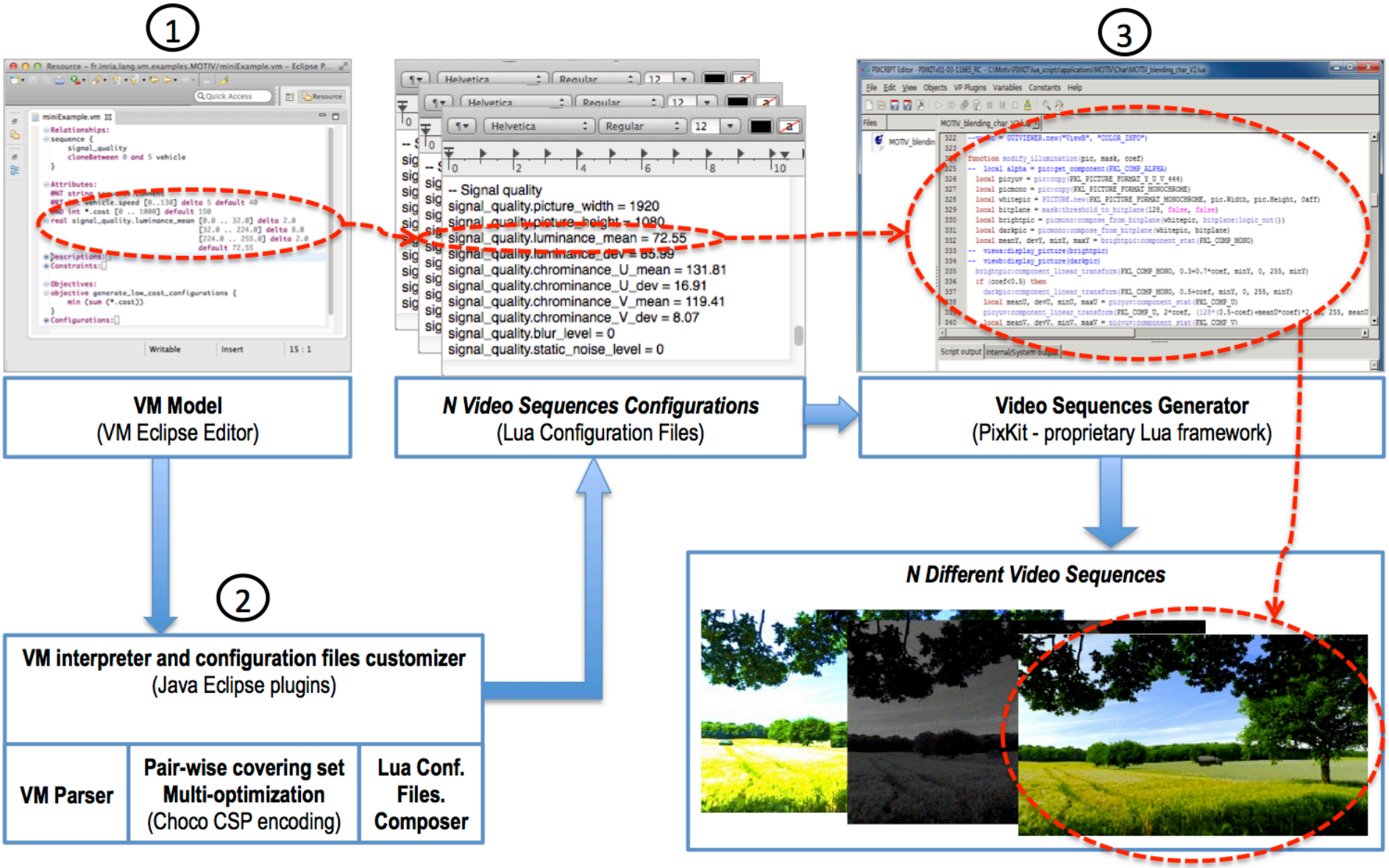
✗ ✗



Software Generator

(derivation engine)


Old Approach
Developers
modify
N times
Video Sequences Generator
N Configuration Files
...
generates
N Video
Sequences
...
VM Approach
Developers and
Domain Experts
model
1 time
generates
VM interpreter, and
configuration files
customizer
VM model
Artifact(s)
Processing
tool(s)


1
3
-- Signal quality
signal_quality.picture_width = 1920
signal_quality.picture_height = 1080
signal_quality.luminance_mean = 72.55
signal_quality.luminance_dev = 55.99
signal_quality.chrominance_U_mean = 131.81
signal_quality.chrominance_U_dev = 16.91
signal_quality.chrominance_V_mean = 119.41
signal_quality.chrominance_V_dev = 8.07
signal_quality.blur_level = 0
signal_quality.static_noise_level = 0
VM Model
(VM Eclipse Editor)
N Video Sequences Configurations
(Lua Configuration Files)
Video Sequences Generator
(PixKit - proprietary Lua framework)
2
VM interpreter and configuration files customizer
(Java Eclipse plugins)
VM Parser
Pair-wise covering set Multi-optimization
(Choco CSP encoding)
Lua Conf. Files. Composer
N Different Video Sequences

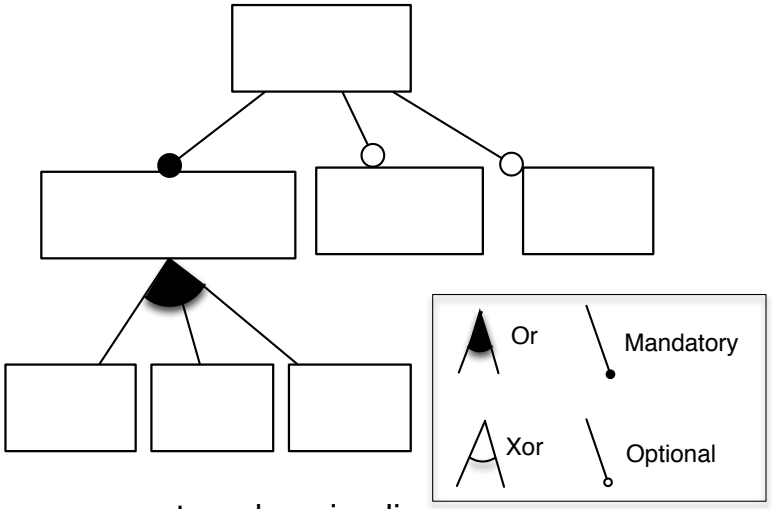
(configuration file)

```
-- 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")
```

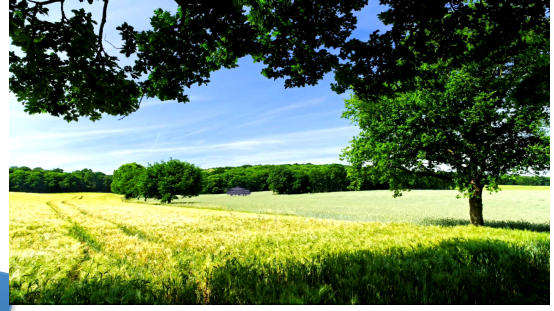
(Lua code)

We synthesize video sequence variants with variability techniques



not, and, or, implies

Variability Models (feature models)



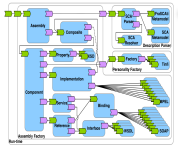
Take Away Messages

- Variability is everywhere
  - Like software is everywhere
  - Numerous artefacts, domains, kinds of systems are subject to customization
- Engineering variability can be hard
  - Exponential number of products
  - Basic or sophisticated techniques (than span your entire cursus) exist
- You're now aware of that!
  - #e1 Recognize variability
  - #e2 Re-visit your cursus and body of knowledge under the angle of variability
  - #e3 Apply state-of-the-art techniques

Reverse Engineering

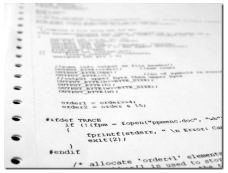
Compose/Decompose
Configure
Analyse
Generate

Component Models



Dependencies Files

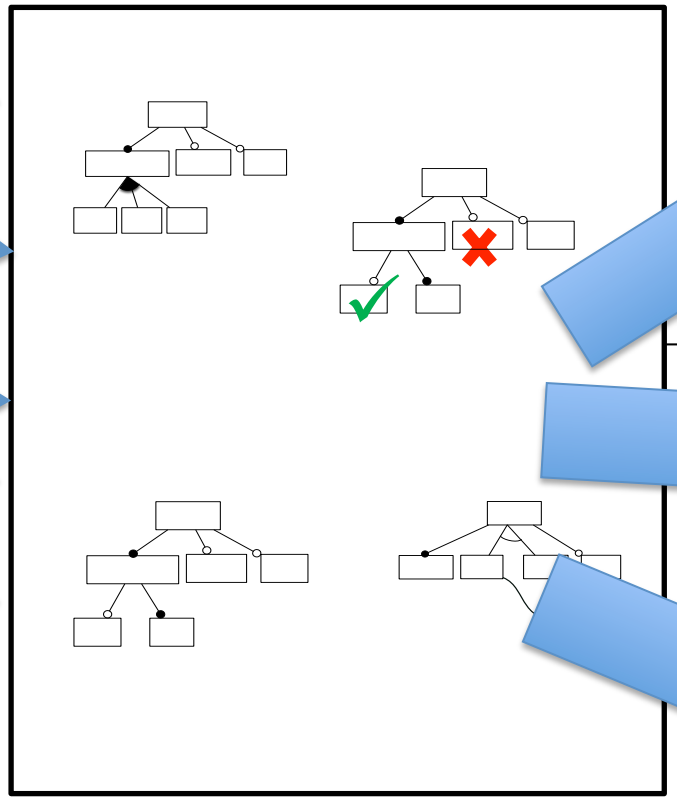
Source Code



Product descriptions

Regulatory Requirements

Web Configurators



FAMiliAR

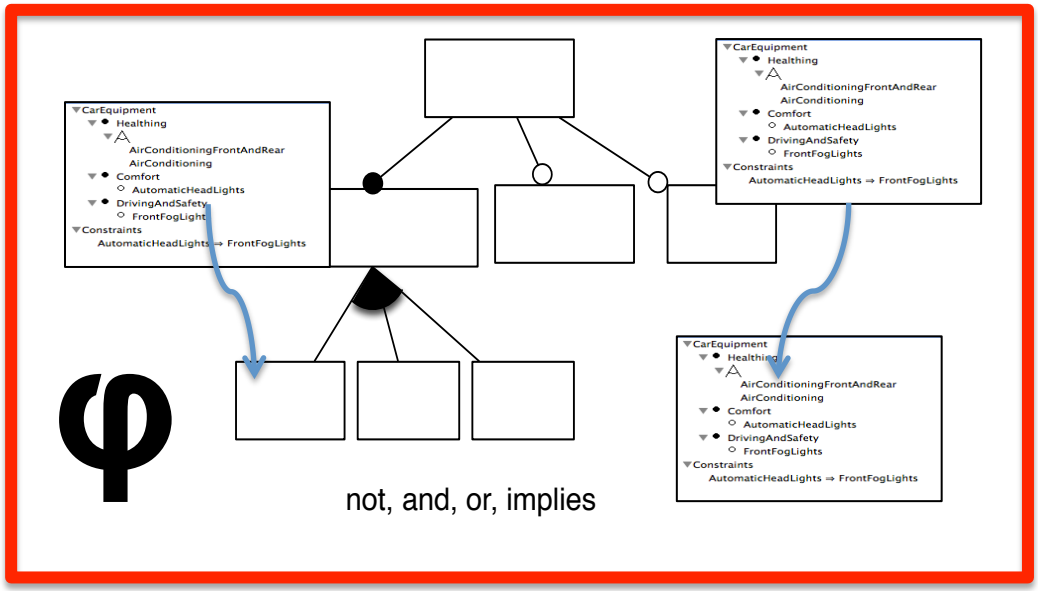
Applications: Softpedia, 100+ web configurators, video analysis, video protection, medical imaging, Wikipedia, plugin architectures, Linux, etc.

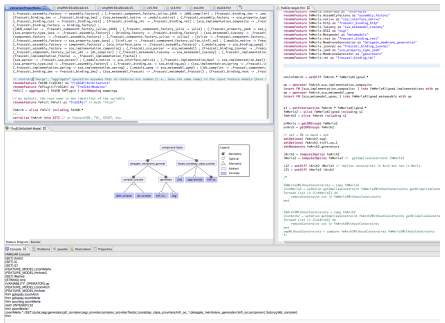
FAMiliAR

(FeAture Model scrIpt Language for manIpulation and Automatic Reasoning)

http://familiar-project.github.com/

COMMON VARIABILITY LANGUAGE





« Fluent » API (Java, for an integration into JVM-based applications)
Dedicated language and environment (Web, Eclipse)

Applications: Softpedia, 100+ web configurators, video analysis, video protection, medical imaging, Wikipedia, architectures à plugins, Linux, etc.