Software Product Lines and Requirements

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

Material

http://mathieuacher.com/teaching/IDE1718/

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, requirements
- Feature Models
  - Defacto standard for modeling product lines and variability requirements
  - Syntax, semantics, automated reasoning

What you will learn

- 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 requirement techniques
- Feature modeling
  - A widely used formalism for modeling product lines and configurable systems in a broad sense (incl. for requirements)

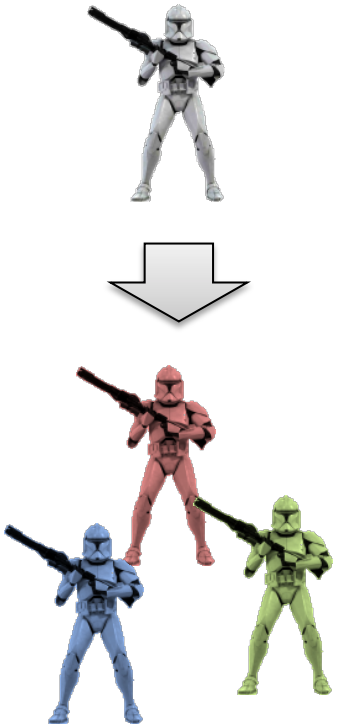
Product Lines

(Software) Product Lines

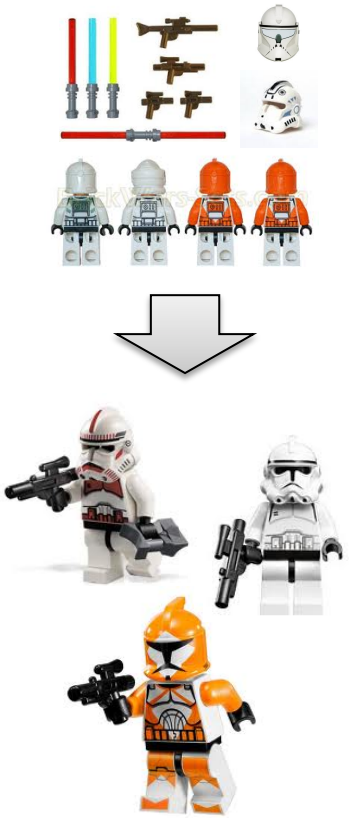
The three ways to build a (software) product

Indepently

„Clone & Own“



„Shared“ (reusable) Assets



(credits: Thorsten Berger’s slide)

The three ways to build a (software) product

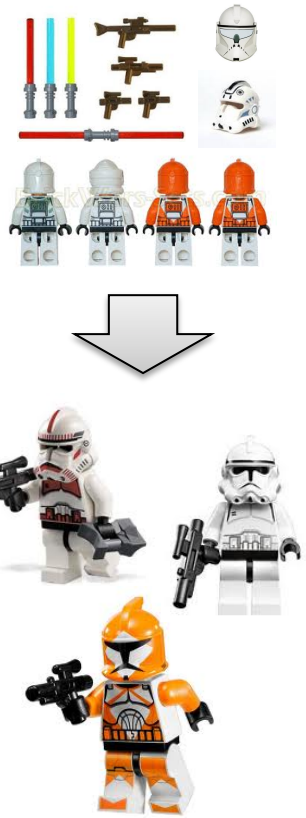
Indepently

„Clone & Own“

„Shared“ (reusable) Assets

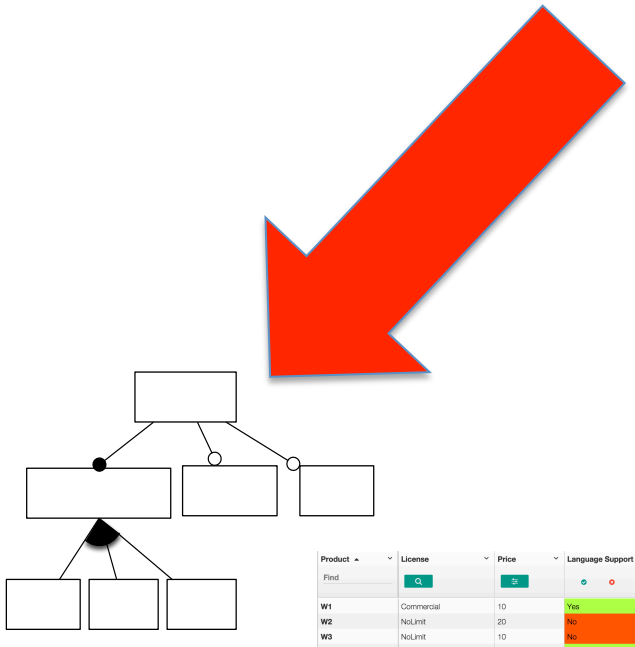
Software Product Lines

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



(credits: Thorsten Berger’s slide)

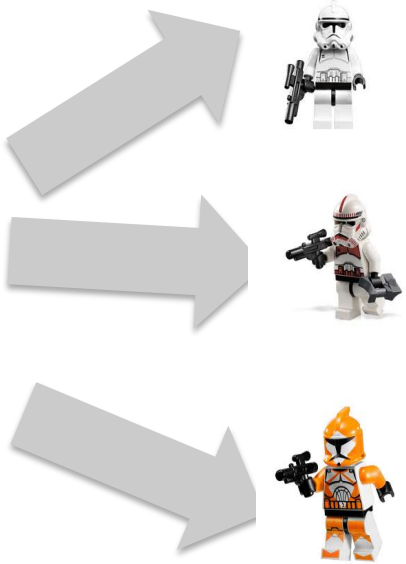
Modeling Variability



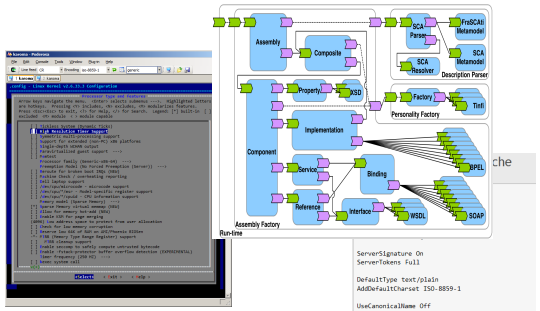
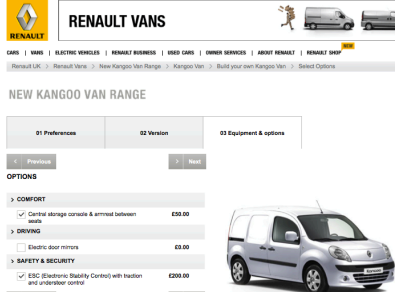
Product	License	Price	Language Support	Language	WYSIWIG
	Find				
W1	Commercial	10	Yes	Java	Yes
W2	NoLimit	20	No		Yes
W3	NoLimit	10	No		Yes
W4	GPL	0	Yes	Python	Yes
W5	GPL	0	Yes	Perl	Yes
W6	GPL	10	Yes	Perl	Yes
W7	GPL	0	Yes	PHP	No
W8	GPL	10	Yes	PHP	Yes

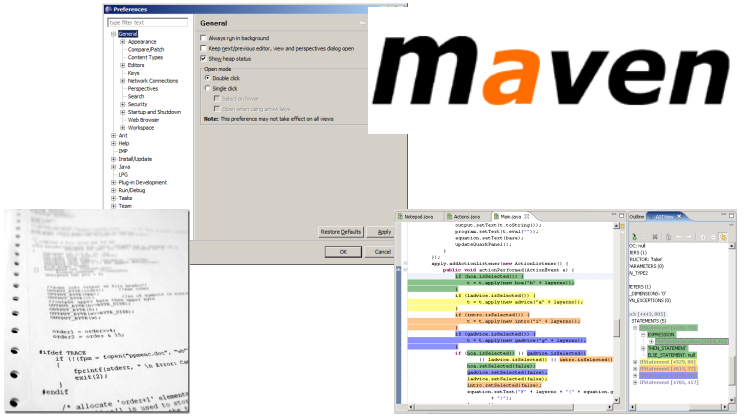
Feature models
or Product Matrices

(product lines)



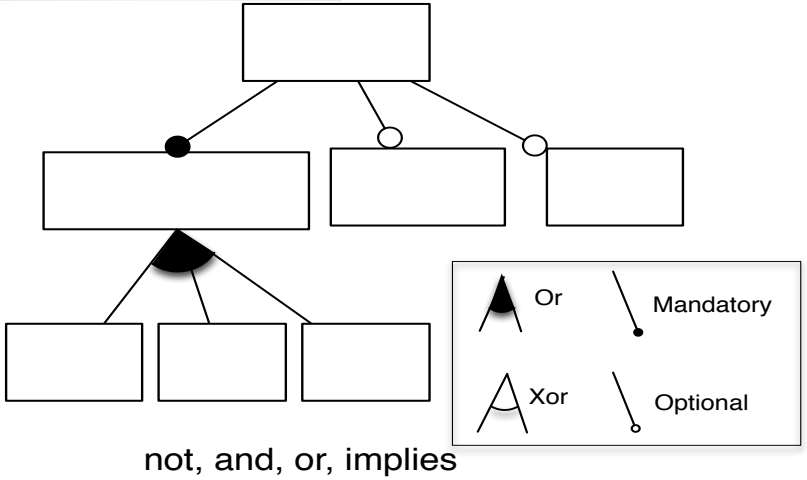
Variants of code (e.g., Java or C)
Variants of user interfaces
Variants of video sequences
Variants of models (e.g., UML or SysML)
Variants of « things » (3D models)
…

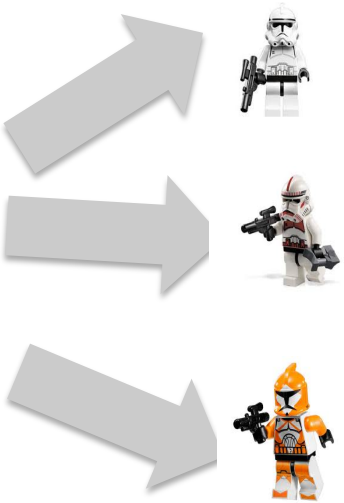


Analyzing/Extracting
Encoding/Formalizing

FAMiliAR

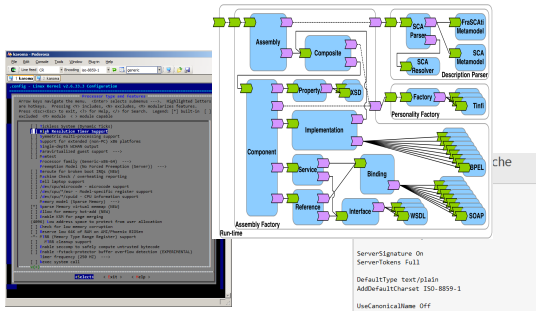
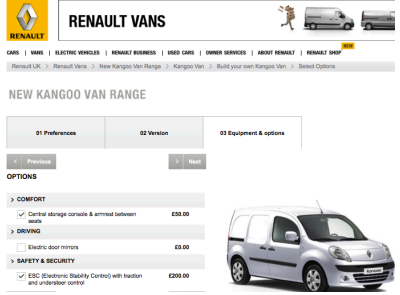


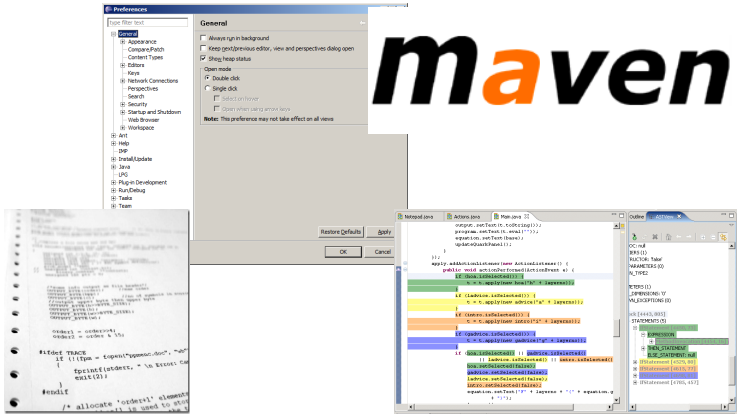
not, and, or, implies



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)
…

Variability Models (feature models)



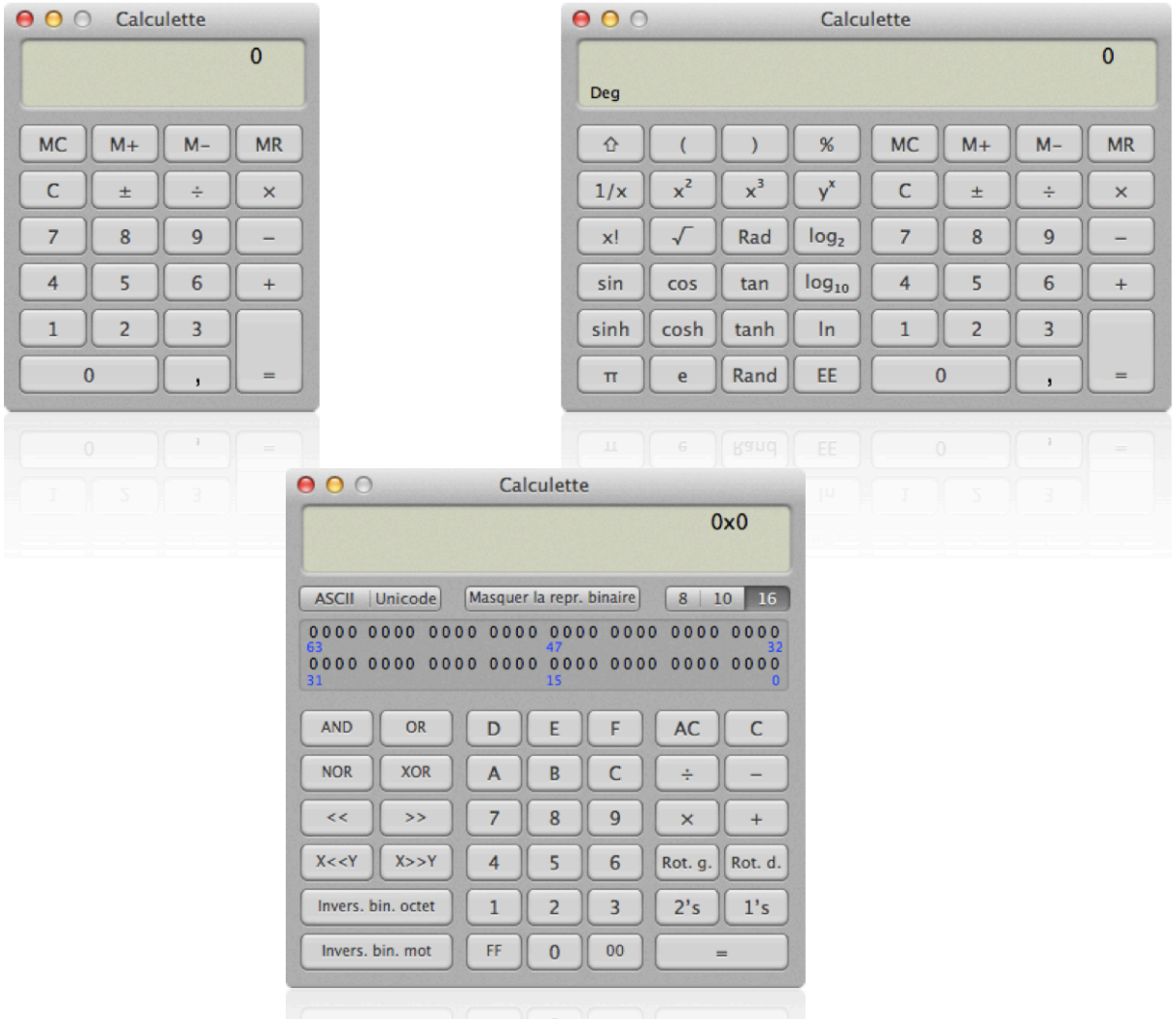
Analyzing/Extracting
Encoding/Formalizing

OpenCompare

Product	License	Price	Language Support	Language	WYSIWIG
Find					
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


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)

NOKIA
Connecting People

Know our past. Create the future...

1982 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994

1995 1996 1997 1998 1999

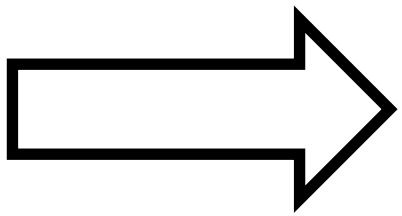
2000 2001 2002

2003 2004

2005

2006 UI

Software-intensive systems



come in many **variants**

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**

Firefox

about:config

Firefox | about:config

Search:

Preference Name	Status	Type	Value
alerts.disableSlidingEffect	default	boolean	false
app.feedback.baseURL	default	string	https://input.mozi
app.support.baseURL	default	string	https://support.m
app.update.altwindowtype	default	string	Browser:About
app.update.auto	default	boolean	false
app.update.autoInstallEnabled	default	boolean	false
app.update.backgroundMaxErrors	default	integer	10
app.update.badge	default	boolean	false
app.update.cert.checkAttributes	default	boolean	true
app.update.cert.maxErrors	default	integer	5
app.update.cert.requireBuiltIn	default	boolean	true
app.update.certs.1.commonName	default	string	aus4.mozilla.org

MAISO

LE PLIAGE PERSONNALISÉ

LE PLIAGE CUIR | LE PLIAGE TOILE

MODÈLES

- Porte-monnaie Toile
- Pochette Toile
- Sac Taille 1 Toile
- Sac Taille 2 Toile
- Sac Taille 3 Toile
- Sac Taille 4 Toile

COULEUR RECTO

COULEUR VERSO

BOUCLERIE

RESET

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

- 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
- Screensavers
- Security
- Spotlight Plugins & Utilities
- System Utilities
- Utilities
- Video
- Word Processing

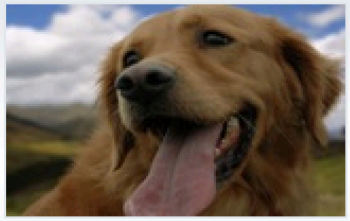
LOBAL PAGES >>
EWS ARCHIVE >>
OFTPEDIA REVIEWS >>
EET THE EDITORS >>

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 >]

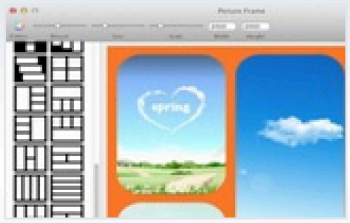
Size:	102 KB
Platform:	Mac OS X 10.8 or later
License:	Commercialware
Rating:	NOT RATED
Downloads:	21
Updated:	June 20th, 07:56 UTC

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

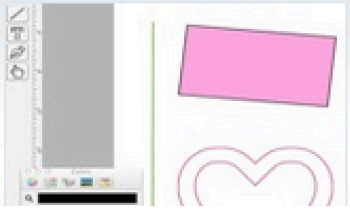


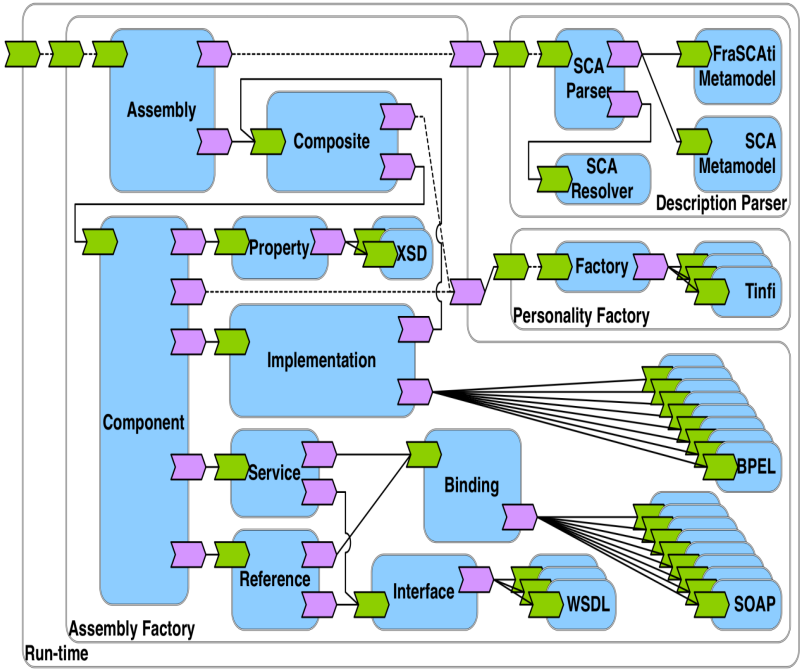
FashionLab Studio 1.1 update

Makes it easy to design your own T-shirt using a Mac

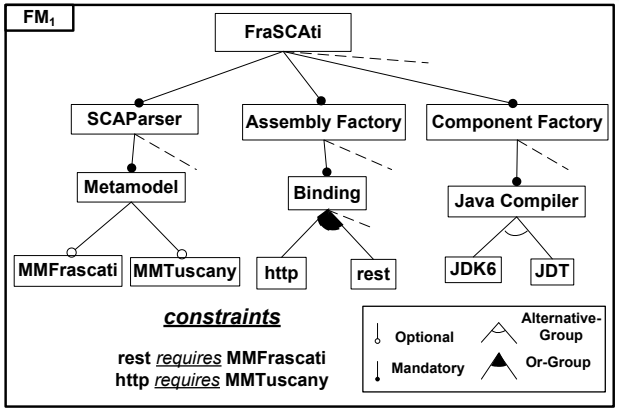
[read more >]

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



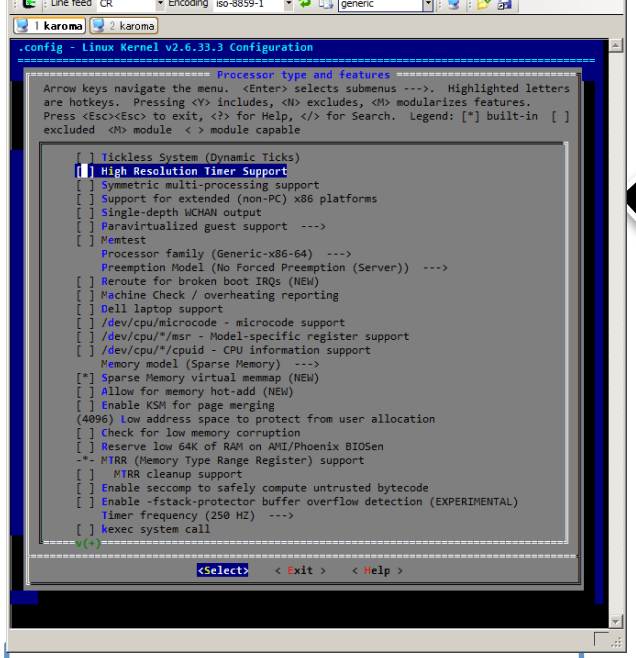


maven



Variability Model

Printer Firmware



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 perfor-mance[1]
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

WIKIPEDIA
The Free Encyclopedia

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

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

 _     _   ___   __  _____  ____  ___       __  _____   __    __    _
| |_| | | | |_) ( (`  | |  | |_  | |_)     ( (`  | |   / /\  / /`  | |_/
|_| | |_| |_|   _)_)  |_|  |_|__ |_| \     _)_)  |_|  /_/--\ \_\_, |_| \
                              ____  ___   ___
                             | |_  / / \ | |_)
                             |_|   \_\_/ |_| \
              _    __    _       __       ___   ____  _     __
             | |  / /\  \ \  /  / /\      | | \ | |_  \ \  / ( (`
           \_|_| /_/--\  \_\/  /_/--\     |_|_/ |_|__  \_\/  _)_)

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)
```

Branch: **master**

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

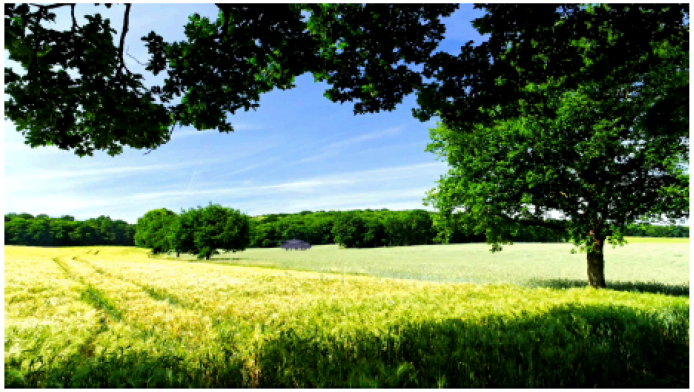
jdubois 2 days ago Use Spring Boot's configuration meta-data

9 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') { %>
```



(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

Video_Sequence_1.cfg

vehicle1.identifier = 15 -- Integer number : 0=disable, 1=AMX30, ..
-- ...
vehicle5.identifier = 12 -- Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0 -- Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 0.80 -- Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level = 1.00 -- Floating point number from 0 (low level) to 1 (high level)
-- *...more attributes*

Video_Sequence_2.cfg

vehicle1.identifier = 7 -- Integer number : 0=disable, 1=AMX30, ..
-- ...
vehicle5.identifier = 3 -- Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0.5 -- Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 0.90 -- Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level = 0.50 -- Floating point number from 0 (low level) to 1 (high level)
-- *...more attributes*

. . .

Video_Sequence_*n*.cfg

vehicle1.identifier = 1 -- Integer number : 0=disable, 1=AMX30, ..
-- ...
vehicle5.identifier = 13 -- Integer number : 0=disable, 1=AMX30, ..
distractors.bird_level = 0 -- Floating point number from 0 (low level) to 1 (high level)
capture.illumination_level = 1.00 -- Floating point number from 0 (low level) to 1 (high level)
signal_quality.blur_level = 0.00 -- Floating point number from 0 (low level) to 1 (high level)
-- *...more attributes*

Video Sequences Generator

Video_Sequence_1.avi

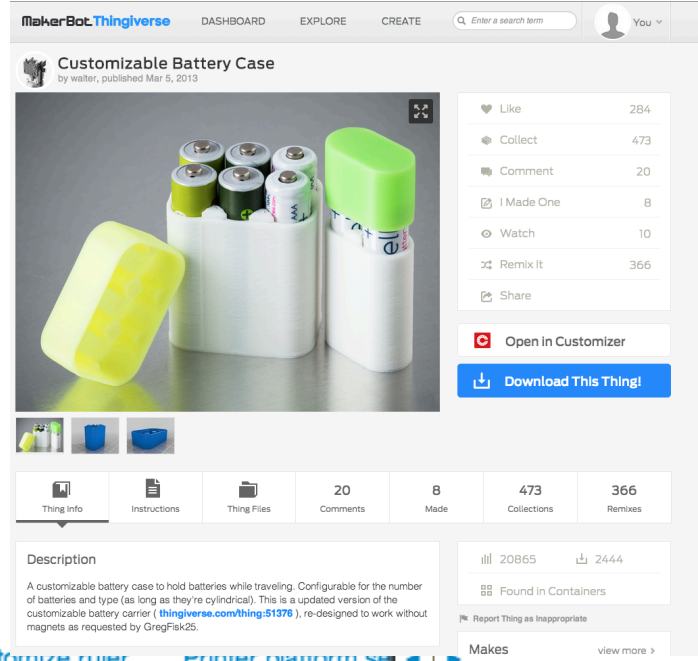
Video_Sequence_2.avi

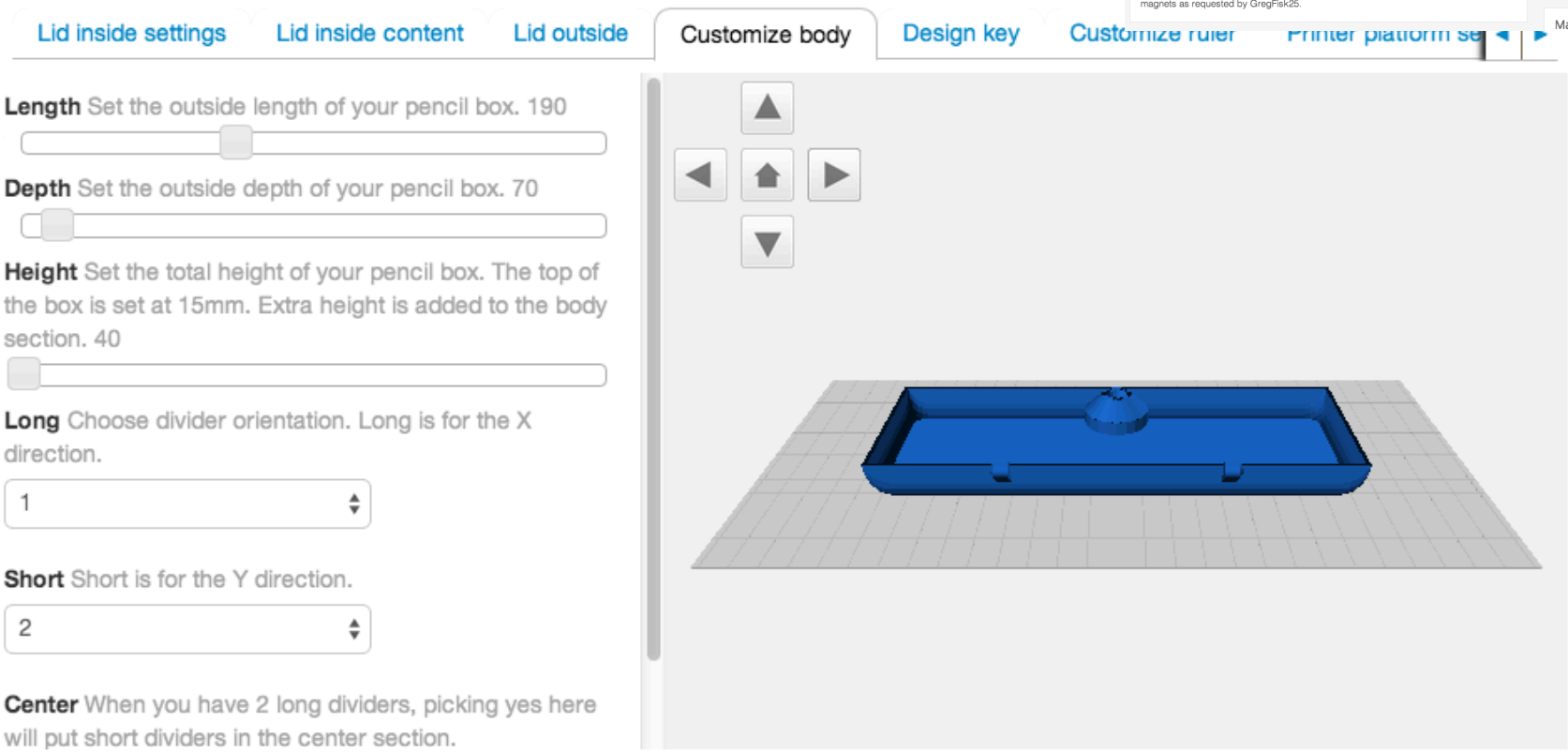
. . .

Video_Sequence_*n*.avi

```
/* [Customize body] */

//Set the outside length of your pencil box.
length=190;//[70:400]
//Set the outside depth of your pencil box.
depth=70;//[50:400]
//Set the total height of your pencil box.  The top of the box is set at 15mm.
//Extra height is added to the body section.
height=40;//[40:150]

//Choose divider orientation.  Long is for the X direction.
long = 1;//[0,1,2]
//Short is for the Y direction.
short = 2;//[0,1,2,3]
//When you have 2 long dividers,
// picking yes here will put short dividers in the center section.
center = 0;//[1:Yes,0:No]
```

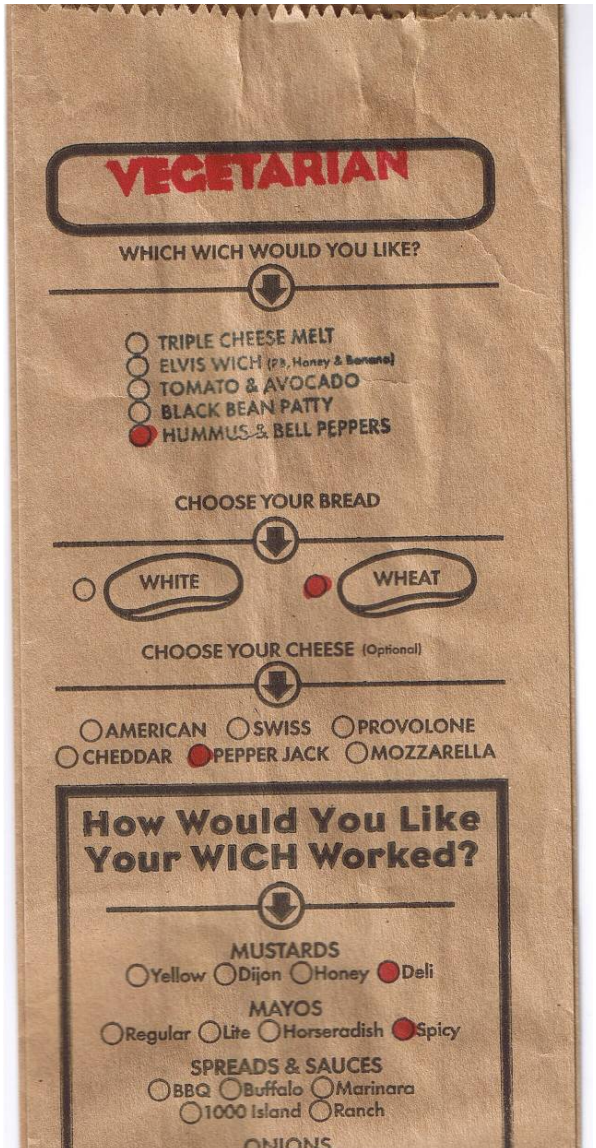
1





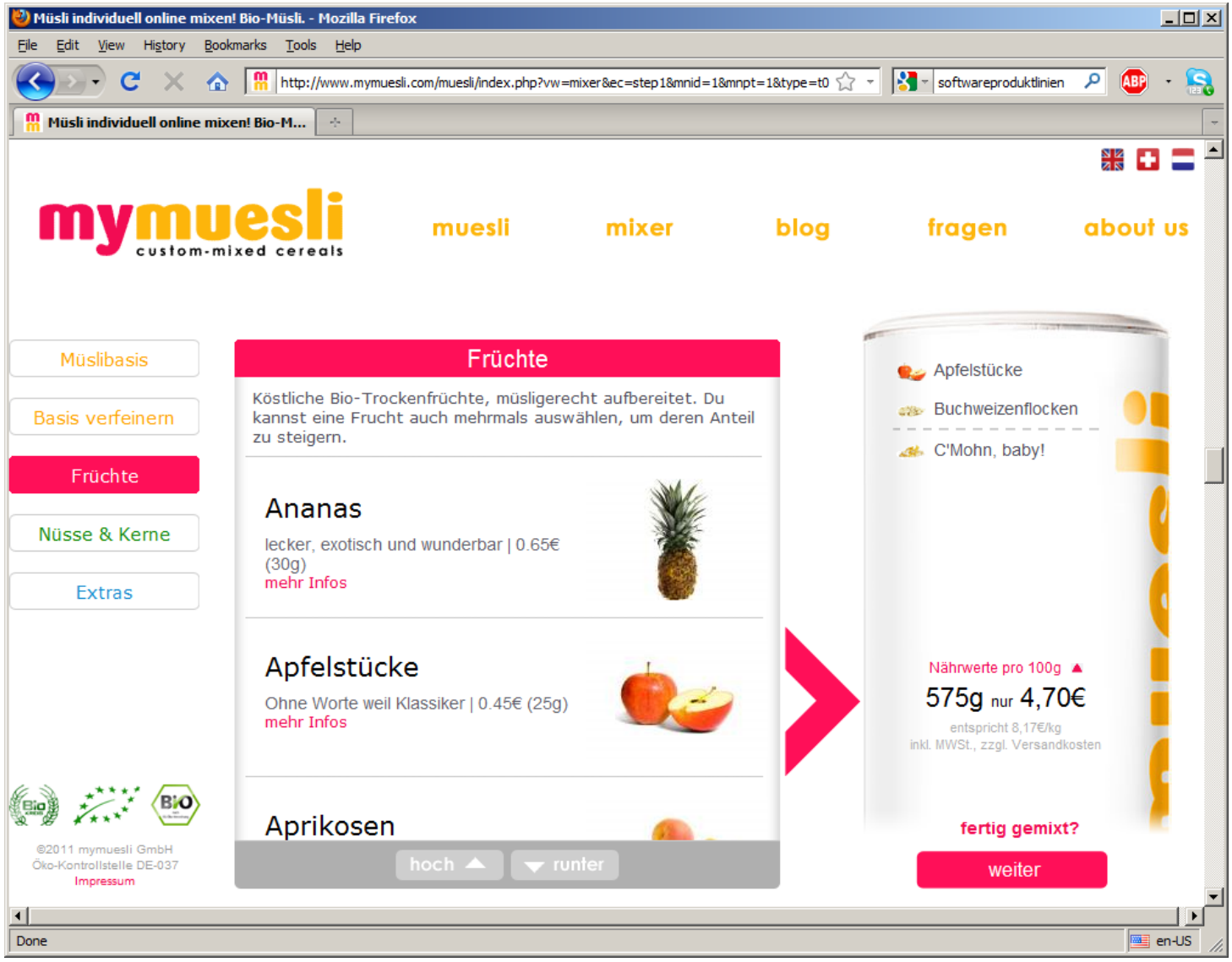
(credits: Christian Kaestner's slide)

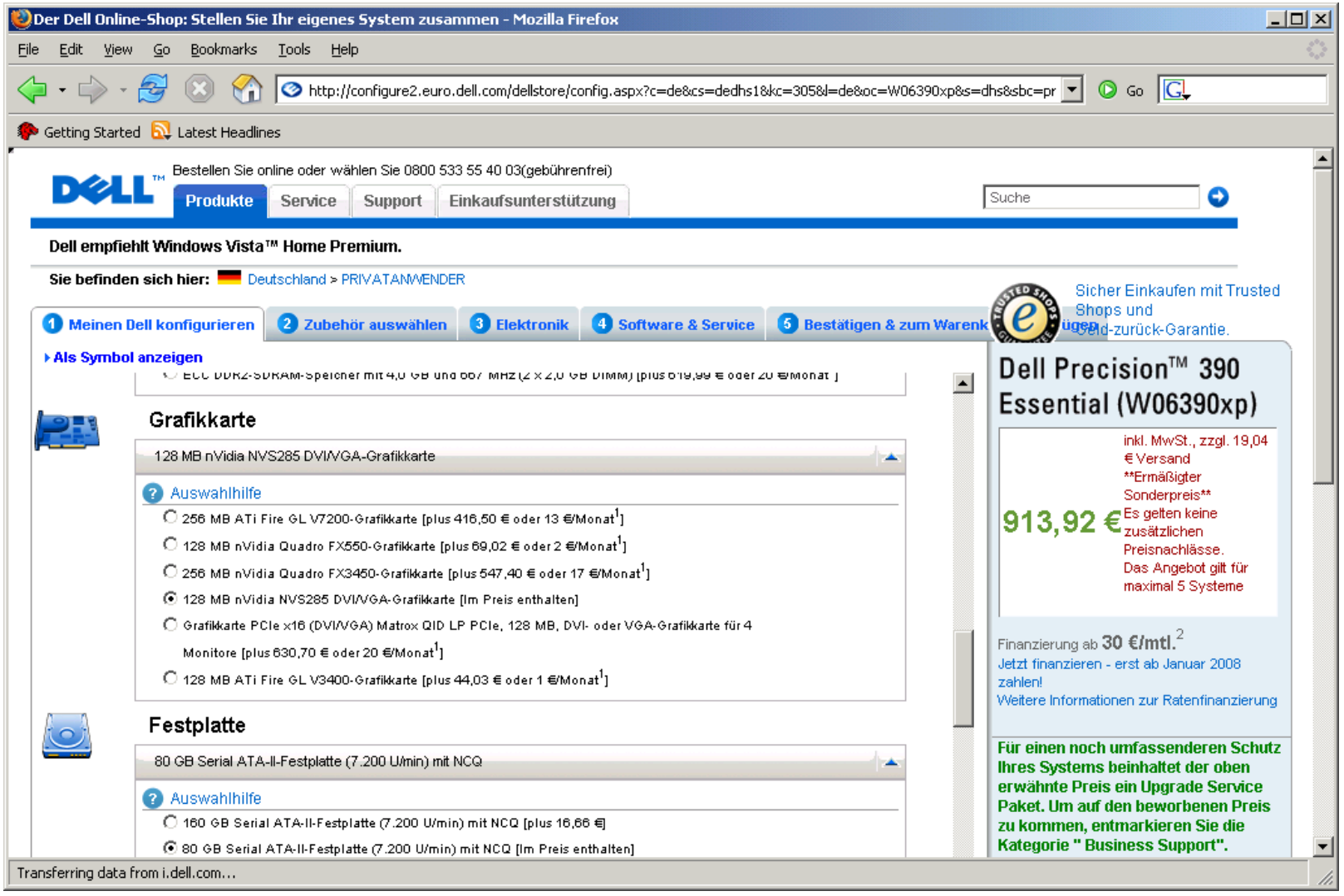
Food? Product lines!



VEGETARIAN
How Would You Like Your WICH Worked?

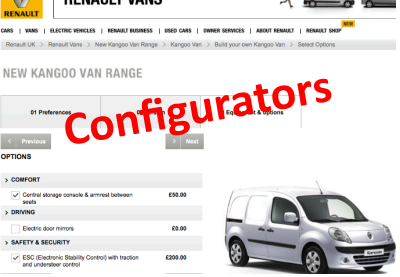
Willkommen bei selve - the shoe individualizer
http://www.selve.net/index_js.html
KOLLEKTION
FUSSTYP
MYSELVE
INFO
HOME
MODELLE
LOOKBOOK
SELVE ID
PASSWORT
>>ANMELDEN
selve
selve Kollektion -> Style: casuals -> Modell: Opal
modell-details
>>hier clicken
>>SELVE SCHUHREGAL
inhalt: 0
>>SHOPPING BAG
inhalt : 0
A. Erstes Oberleder
Velours leder Sand
Velours leder Bordeaux
Velours leder Cognac
Velours leder Sand
B.
C. Futterleder
Beige
D. Absatz
Hufeisen Braun
E. Sohle
Gummisohle
>>ÄNDERN
>>ZURÜCKLEGEN


Müsli individuell online mixen! Bio-Müsli. - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.mymuesli.com/muesli/index.php?vw=mixer&ec=step1&mnid=1&mnpt=1&type=t0
softwareproduktlinien
Müsli individuell online mixen! Bio-M...
mymuesli
custom-mixed cereals
muesli
mixer
blog
fragen
about us
Müslibasis
Basis verfeinern
Früchte
Nüsse & Kerne
Extras
Früchte
Köstliche Bio-Trockenfrüchte, müsligerecht aufbereitet. Du kannst eine Frucht auch mehrmals auswählen, um deren Anteil zu steigern.
Ananas
lecker, exotisch und wunderbar | 0.65€ (30g)
mehr Infos
Apfelstücke
Ohne Worte weil Klassiker | 0.45€ (25g)
mehr Infos
Aprikosen
hoch
runter
Apfelstücke
Buchweizenflocken
C'Mohn, baby!
Nährwerte pro 100g
575g nur 4,70€
entspricht 8,17€/kg
inkl. MWSt., zzgl. Versandkosten
fertig gemixt?
weiter
©2011 mymuesli GmbH
Öko-Kontrollstelle DE-037
Impressum
Done
en-US

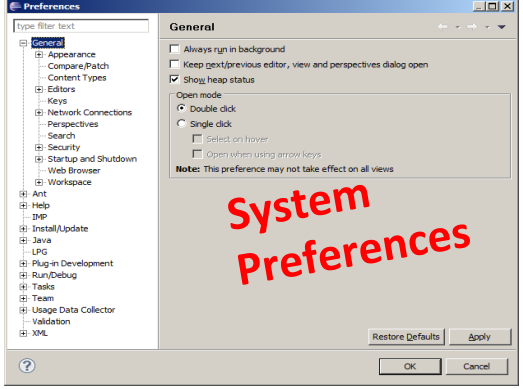

Der Dell Online-Shop: Stellen Sie Ihr eigenes System zusammen - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://configure2.euro.dell.com/dellstore/config.aspx?c=de&cs=dedhs1&kc=305&l=de&oc=W06390xp&s=dhs&sbc=pr
Go
Getting Started
Latest Headlines
Bestellen Sie online oder wählen Sie 0800 533 55 40 03(gebührenfrei)
Produkte
Service
Support
Einkaufsunterstützung
Suche
Dell empfiehlt Windows Vista™ Home Premium.
Sie befinden sich hier: Deutschland > PRIVATANWENDER
1 Meinen Dell konfigurieren
2 Zubehör auswählen
3 Elektronik
4 Software & Service
5 Bestätigen & zum Warenk
Sicher Einkaufen mit Trusted Shops und Geld-zurück-Garantie.
Als Symbol anzeigen
Grafikkarte
128 MB nVidia NVS285 DVI/VGA-Grafikkarte
Auswahlhilfe
256 MB ATi Fire GL V7200-Grafikkarte [plus 416,50 € oder 13 €/Monat1]
128 MB nVidia Quadro FX550-Grafikkarte [plus 69,02 € oder 2 €/Monat1]
256 MB nVidia Quadro FX3450-Grafikkarte [plus 547,40 € oder 17 €/Monat1]
128 MB nVidia NVS285 DVI/VGA-Grafikkarte [Im Preis enthalten]
Grafikkarte PCIe x16 (DVI/VGA) Matrox QID LP PCIe, 128 MB, DVI- oder VGA-Grafikkarte für 4 Monitore [plus 630,70 € oder 20 €/Monat1]
128 MB ATi Fire GL V3400-Grafikkarte [plus 44,03 € oder 1 €/Monat1]
Festplatte
80 GB Serial ATA-II-Festplatte (7.200 U/min) mit NCQ
Auswahlhilfe
160 GB Serial ATA-II-Festplatte (7.200 U/min) mit NCQ [plus 16,66 €]
80 GB Serial ATA-II-Festplatte (7.200 U/min) mit NCQ [Im Preis enthalten]
Dell Precision™ 390 Essential (W06390xp)
913,92 €
inkl. MwSt., zzgl. 19,04 € Versand
Ermäßigter Sonderpreis
Es gelten keine zusätzlichen Preisnachlässe.
Das Angebot gilt für maximal 5 Systeme
Finanzierung ab 30 €/mtl.2
Jetzt finanzieren - erst ab Januar 2008 zahlen!
Weitere Informationen zur Ratenfinanzierung
Für einen noch umfassenderen Schutz Ihres Systems beinhaltet der oben erwähnte Preis ein Upgrade Service Paket. Um auf den beworbenen Preis zu kommen, entmarkieren Sie die Kategorie " Business Support".
Transferring data from i.dell.com...

Product Lines and Variability

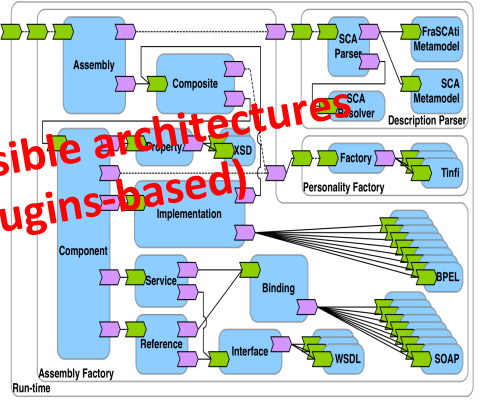
Configurators



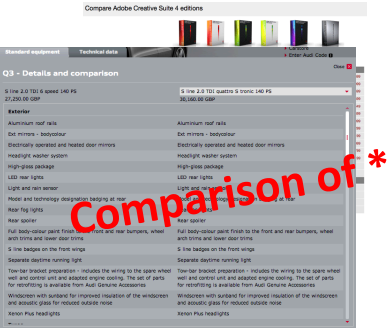
System Preferences



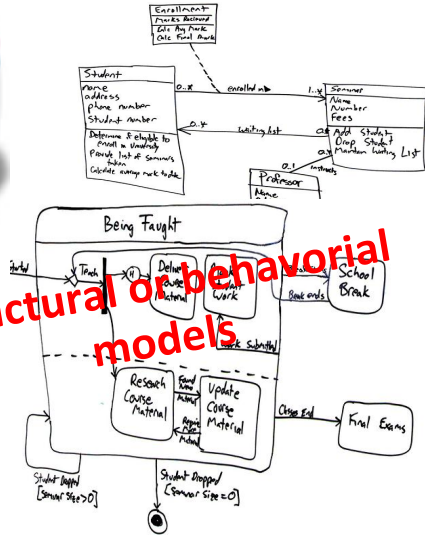
Extensible architectures (eg plugins-based)



Comparison of *



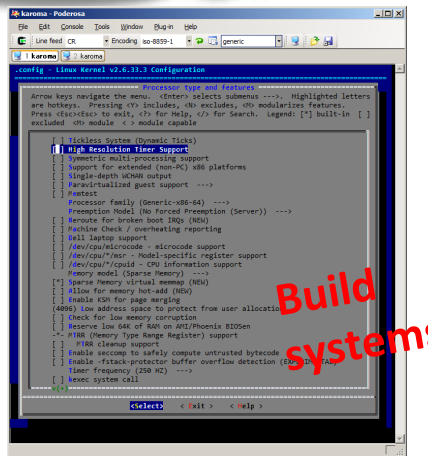
Structural or behavorial models



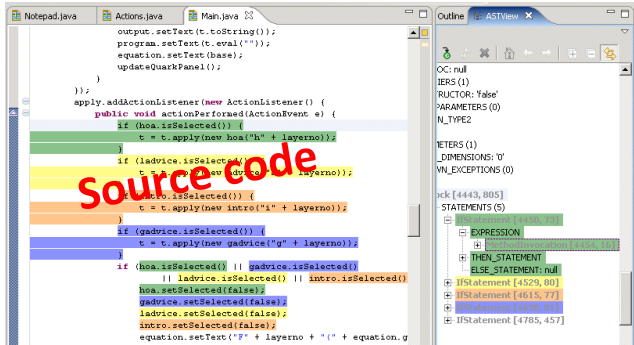
Configuration files

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

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

ServerName localhost:80
ServerAdmin admin@localhost

ServerSignature On
ServerTokens Full

DefaultType text/plain
AddDefaultCharset ISO-8859-1

UseCanonicalName Off

HostnameLookups Off

ErrorLog logs/error.log
LogLevel error

PidFile logs/httpd.pid

Timeout 300

KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15


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

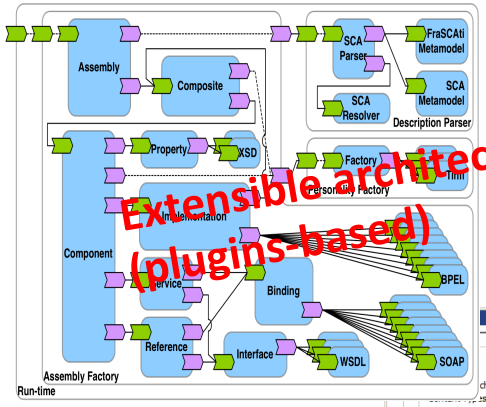
Build systems



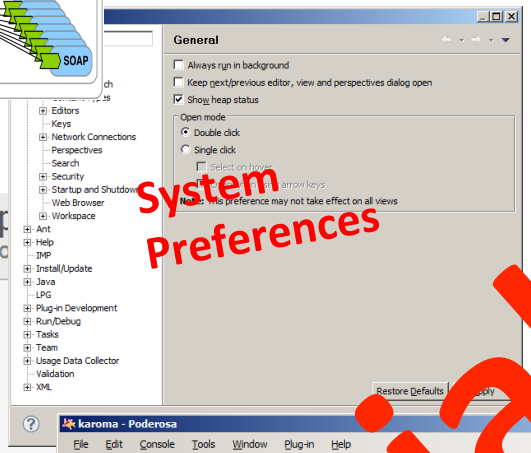
Source code



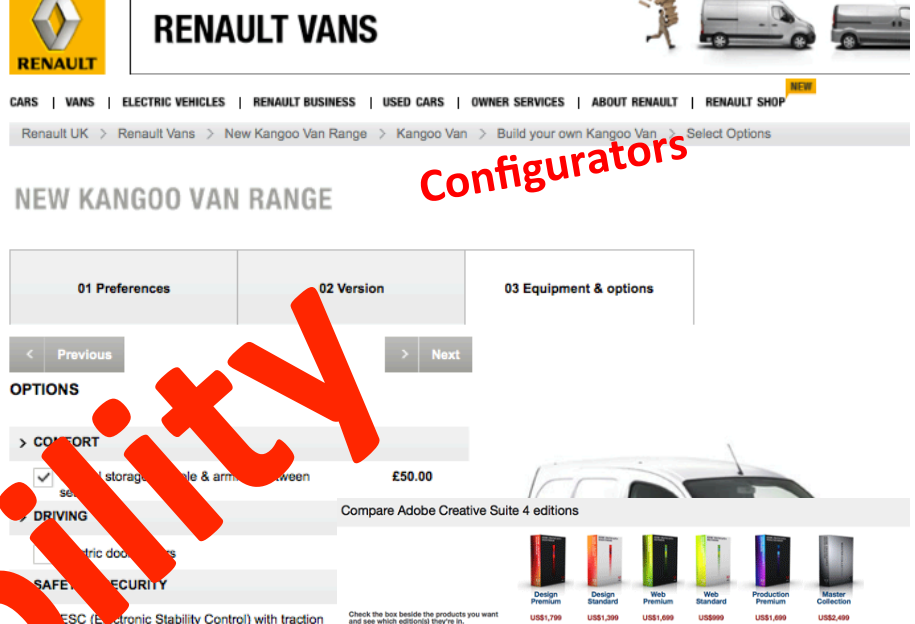
Variability



Extensible architectures (plugins-based)

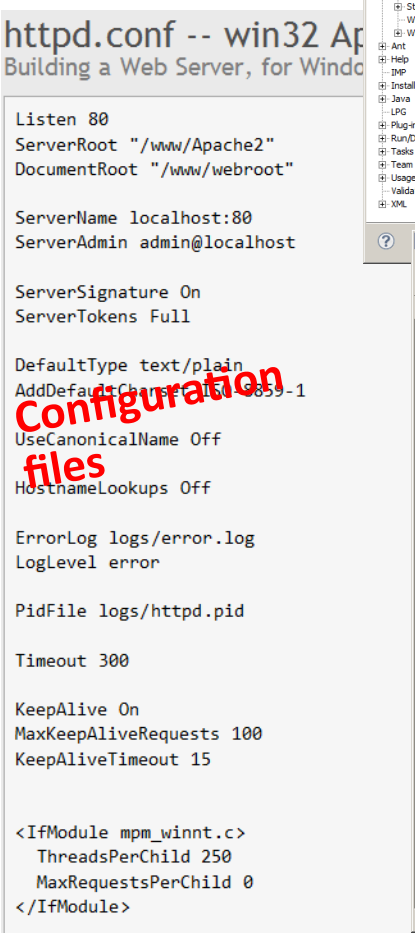


System Preferences

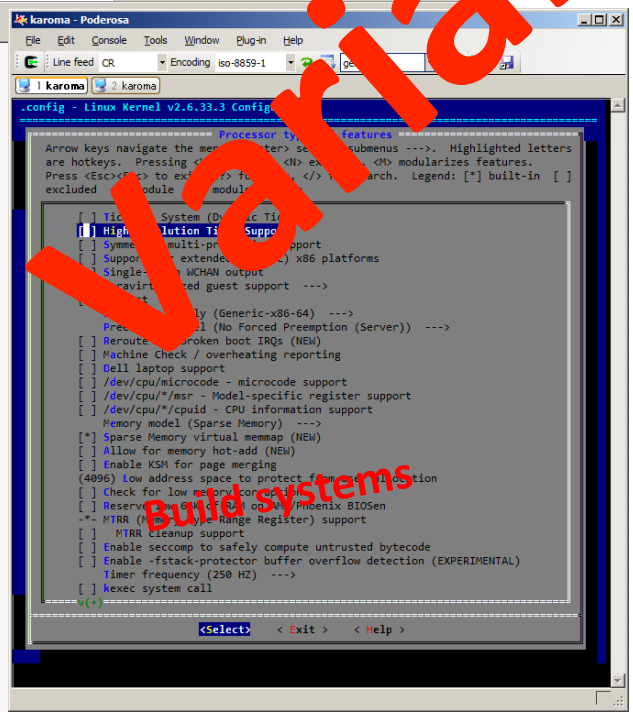


Configurators

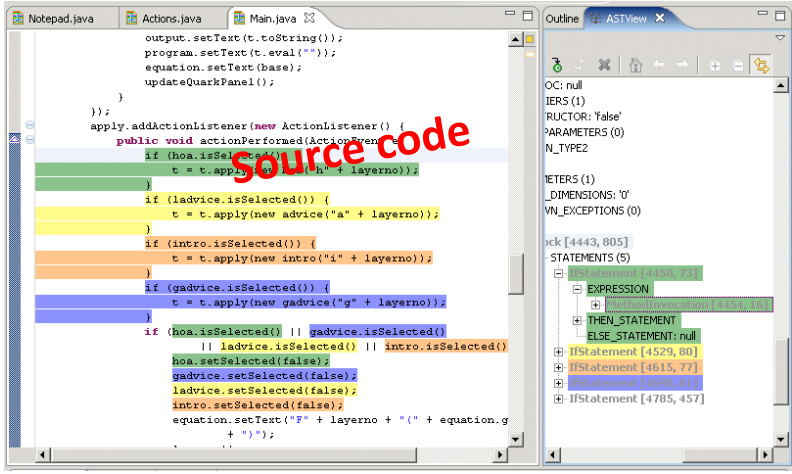
Comparison of Product



Configuration files



Built systems



Source code

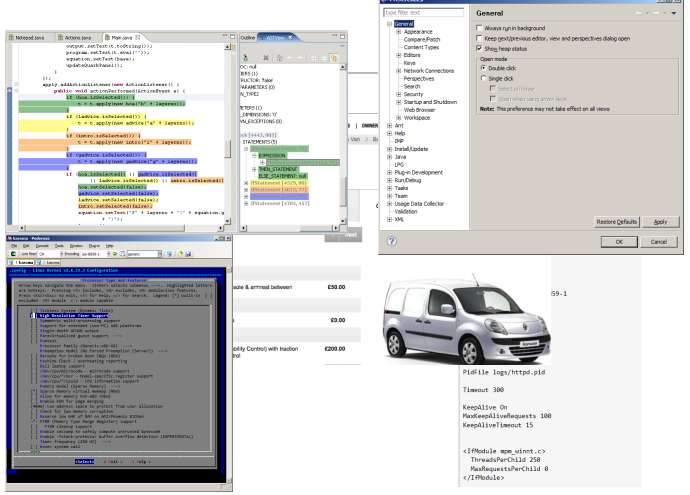
Quizz Time

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

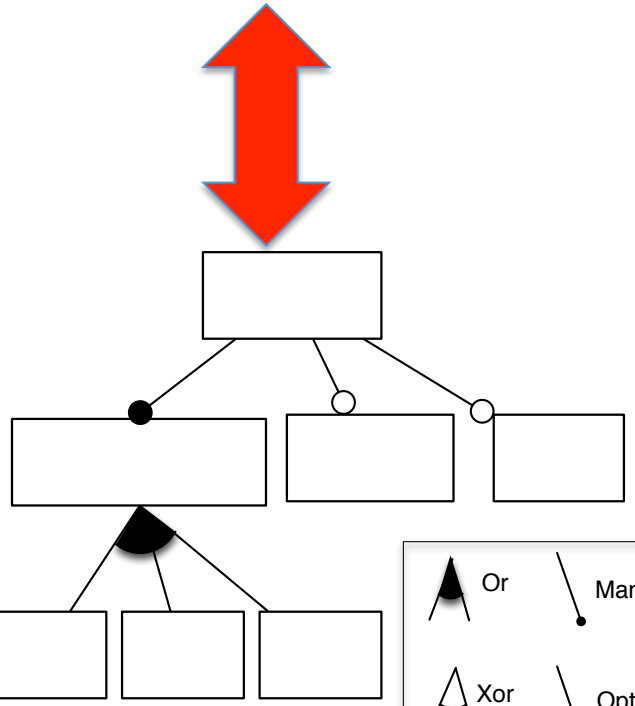
Software is eating the world (any company will be a software company)

With software you can produce variants of software; in fact it is more general: you can produce variants of anything since software is everywhere

If you have the super-power to "**vary**"
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)
…



not, and, or, implies

Variability Models (feature models)

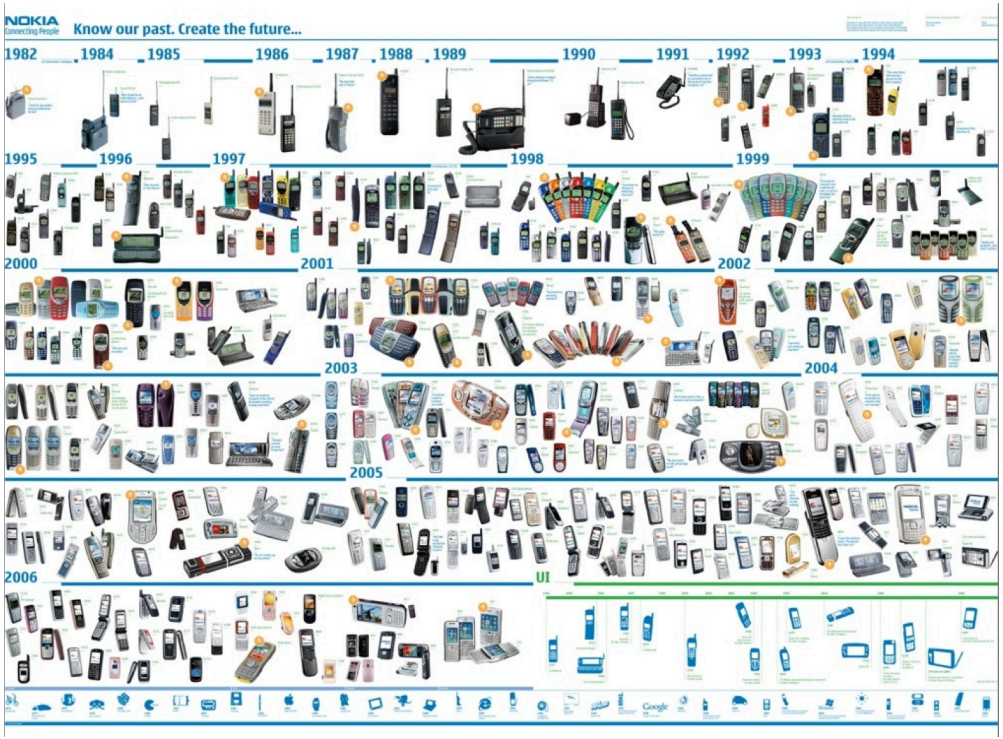
Variability: two definitions

- "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

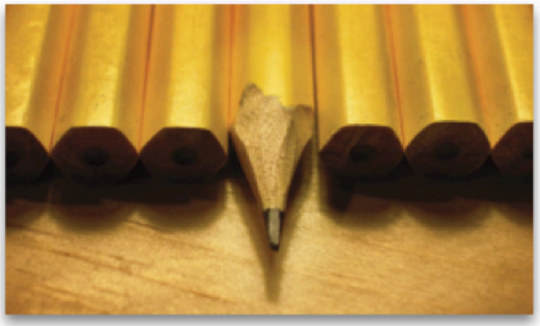


Benefits

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

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%

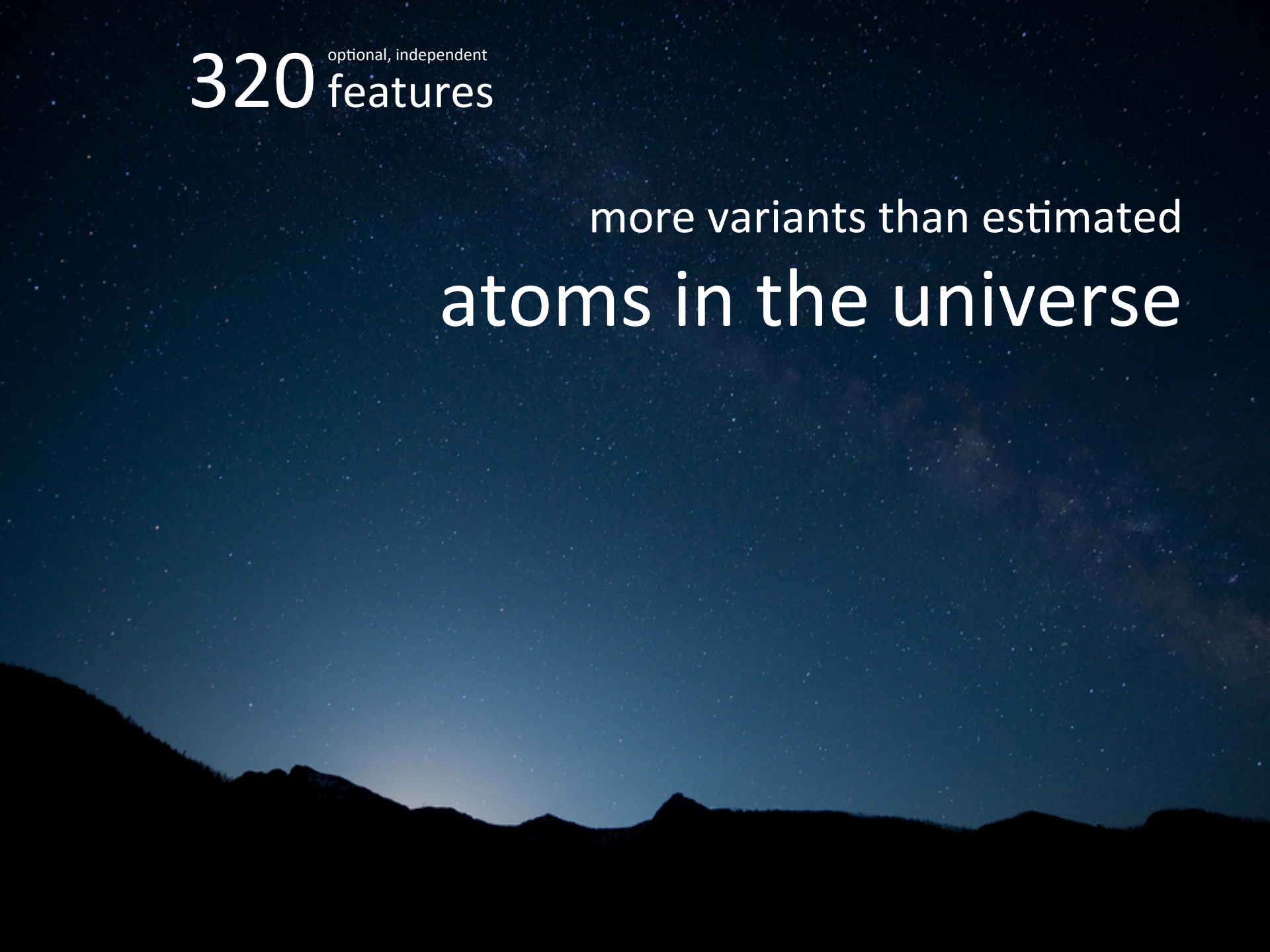
Variability = Complexity

(credits: Christian Kaestner's slide)

33 optional, independent features

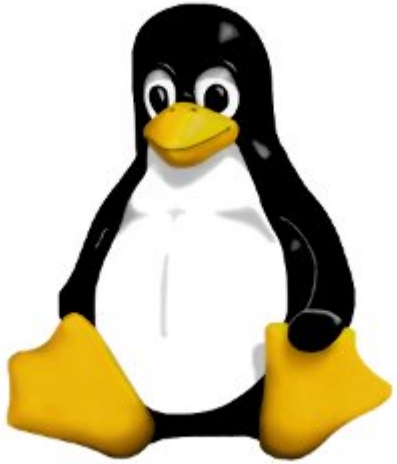
a unique variant for every

person on this planet


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

2000 features

10000 features



The specificity of

Software Product Line Engineering

The development of a

family of software systems

differs from the development of

a **single** software system

THANKS CAPTAIN
OBVIOUS

« The development of a **family** of software systems differs from the development of a **single** software system »

Reuse

Customization

Automation

Commonality

Variability

Assembly Line and Mass Customization

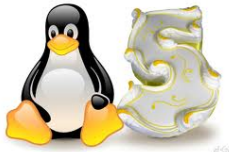
Reuse and Mass Customization

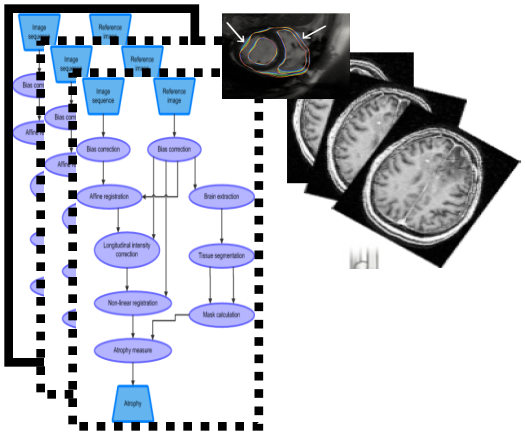
Starting from scratch?

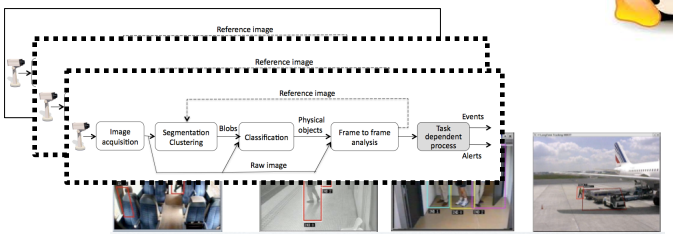
You cannot start from scratch

"a set of software- intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" **[Clements et al., 2001]**

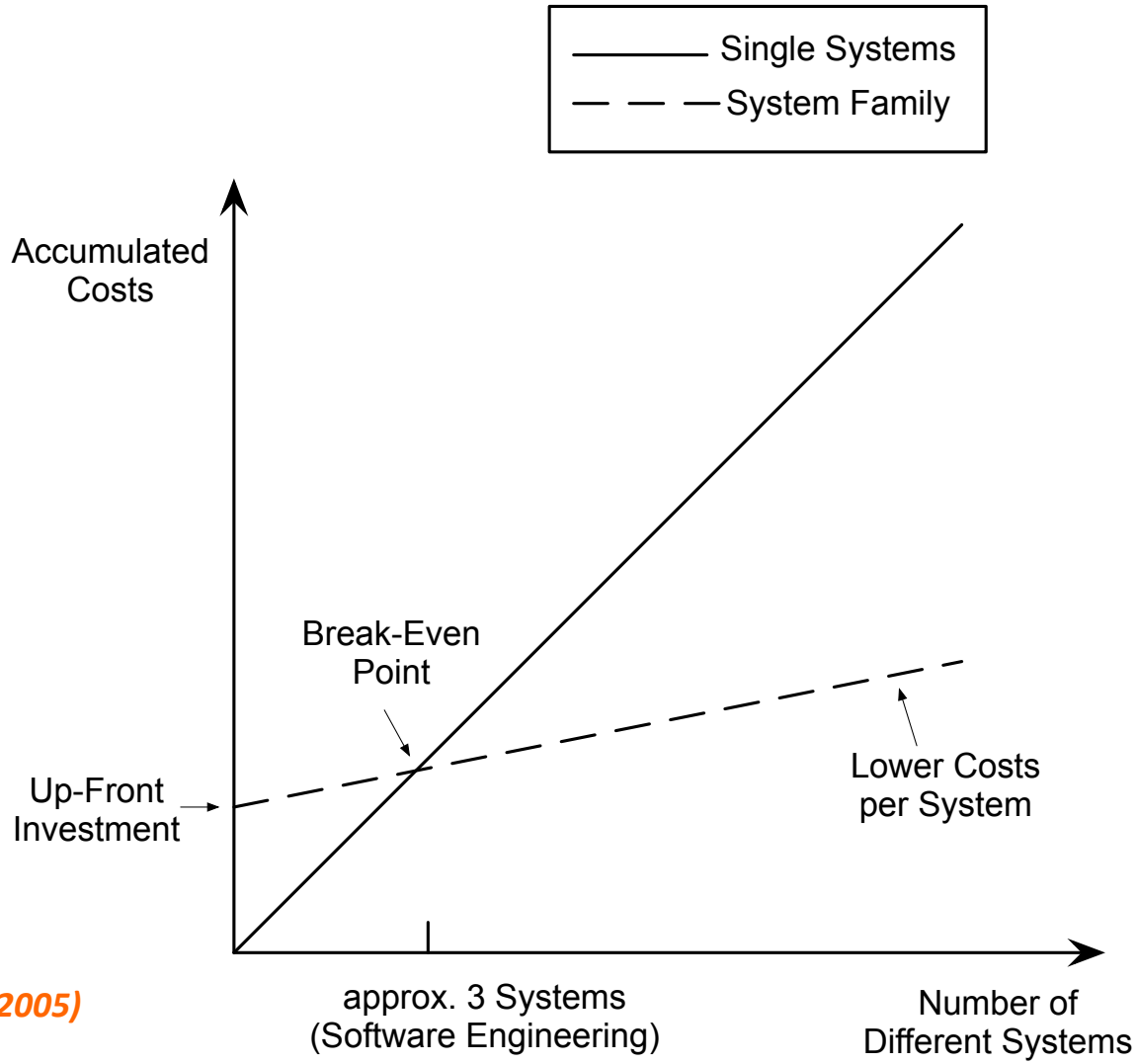
Software Product Lines





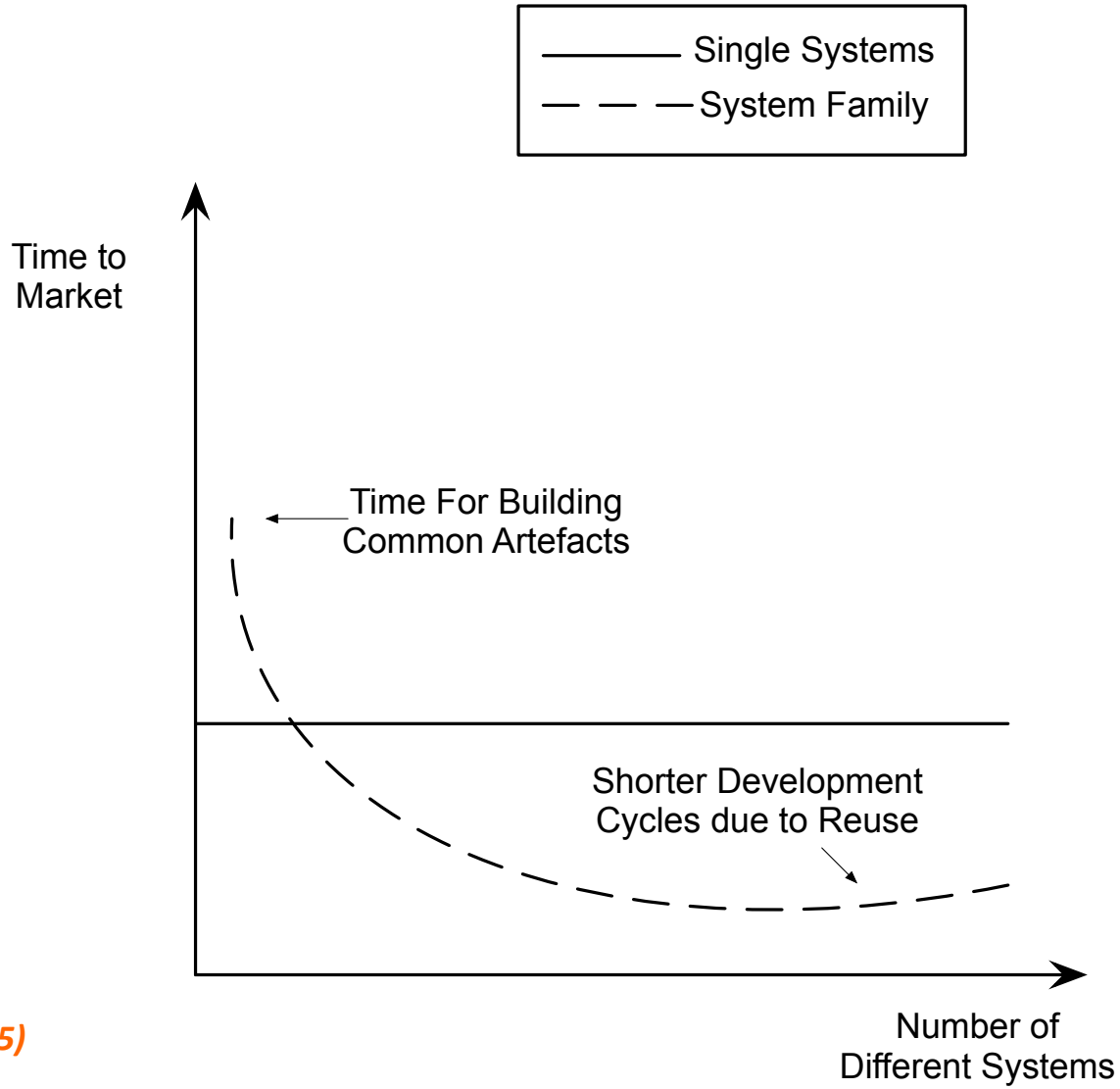


Promises of Software Product Line Engineering



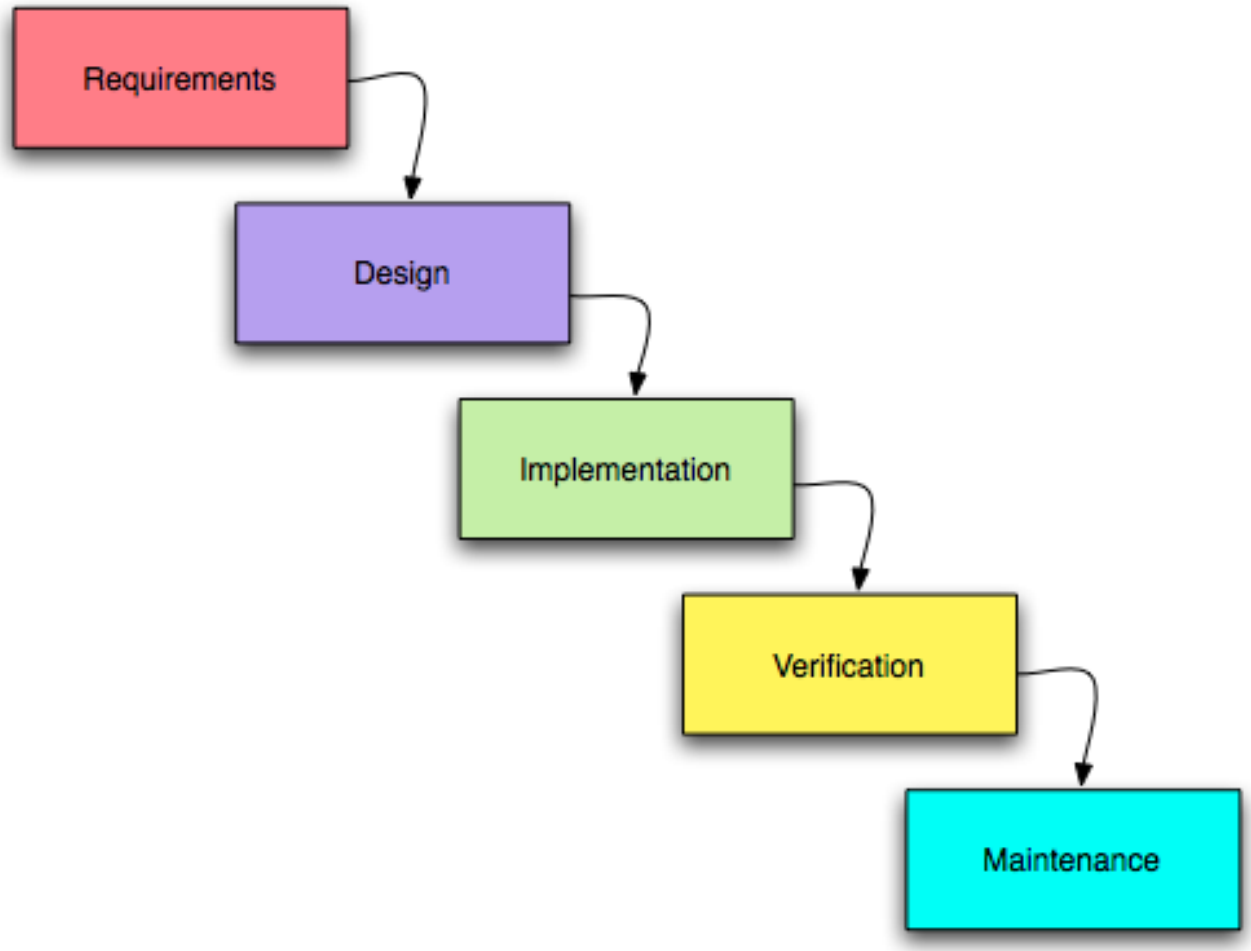
Klaus Pohl (2005)

Promises of Software Product Line Engineering

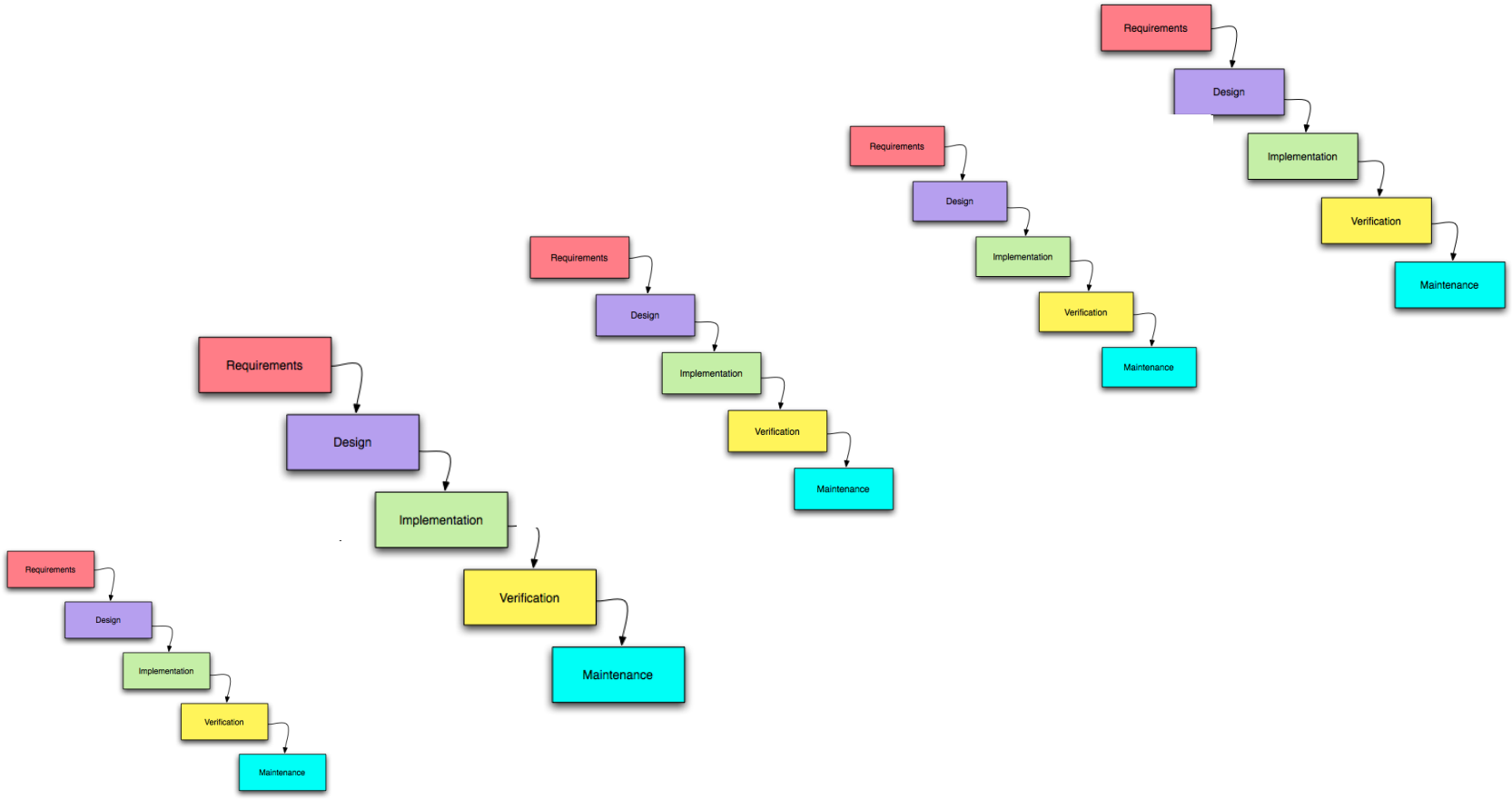


Klaus Pohl (2005)

Single Software Development



Software Product Line Development?



Time and Effort: not scalable!

We need an **engineering process specific** to **software product lines**

Observation: "Reuse-in-the-large works best in families of related systems, and thus is domain dependent." [Glass, 2001]

Domain Engineering

[...] is the activity of collecting, organizing, and storing past experience in building systems [...] in a particular domain in the form of reusable assets [...], as well as providing an adequate means for reusing these assets (i.e., retrieval, qualification, dissemination, adaptation, assembly, and so on) when building new systems.

K. Czarnecki and U. Eisenecker

Domain Engineering

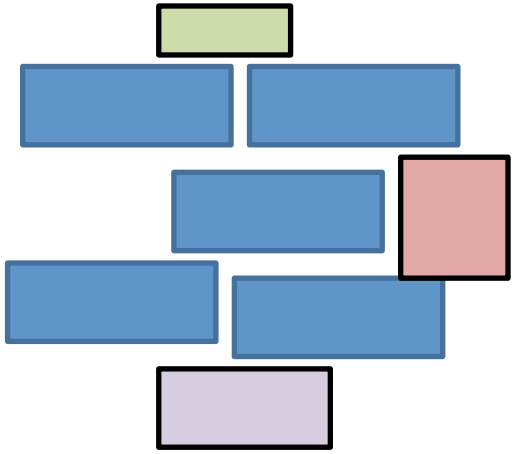
~=

Product Line Engineering

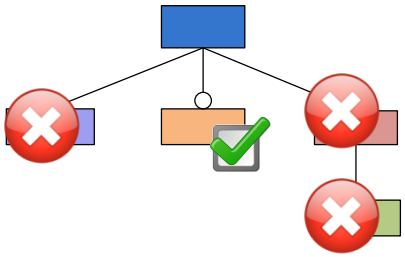
The conventional software engineering concentrates on satisfying the requirements for a **single** system

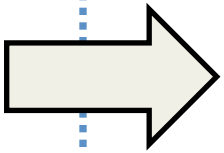
Domain Engineering concentrates on providing **reusable** solutions for **families** of systems.

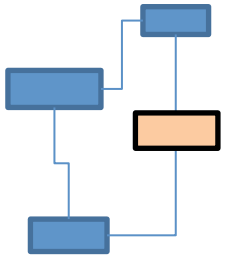
Key idea: building a reusable platform during domain engineering



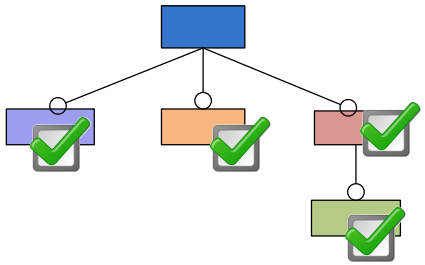
Specific requirements

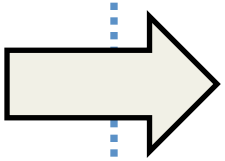


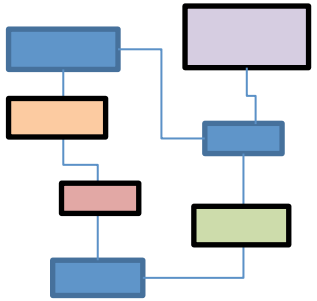




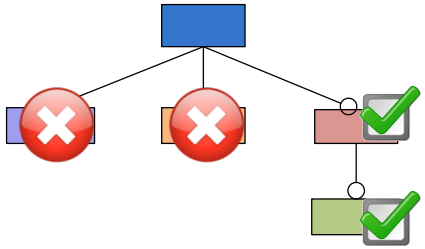
$product_2$

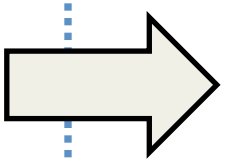


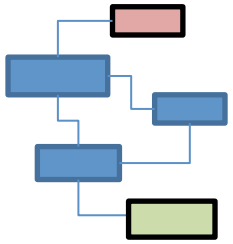




$product_n$





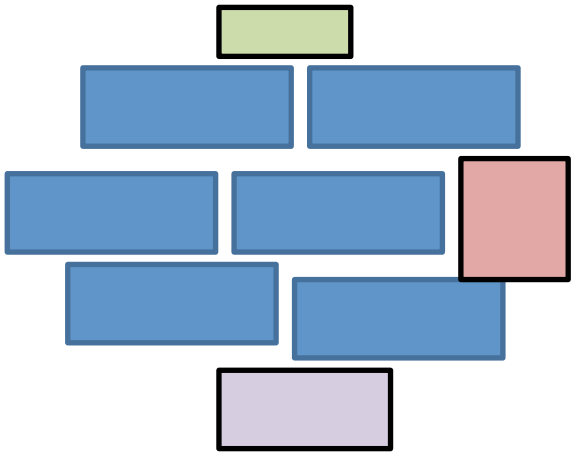


$product_1$

Software Product Line Engineering

Factoring out **commonalities**

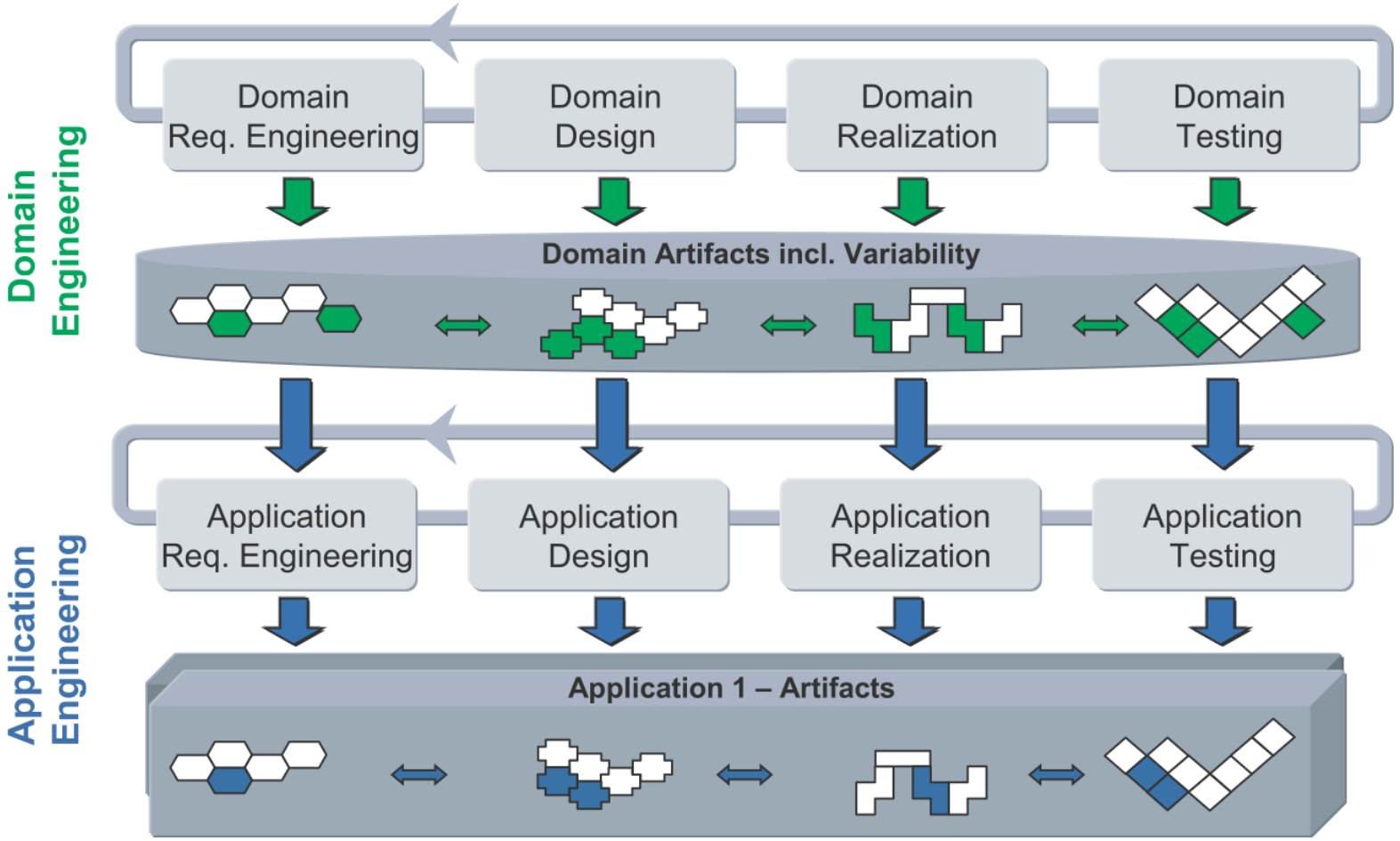
for **Reuse** [Krueger et al., 1992] [Jacobson et al., 1997]



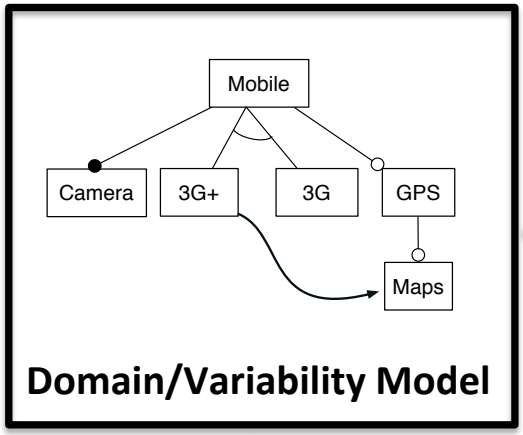
Managing **variabilities**

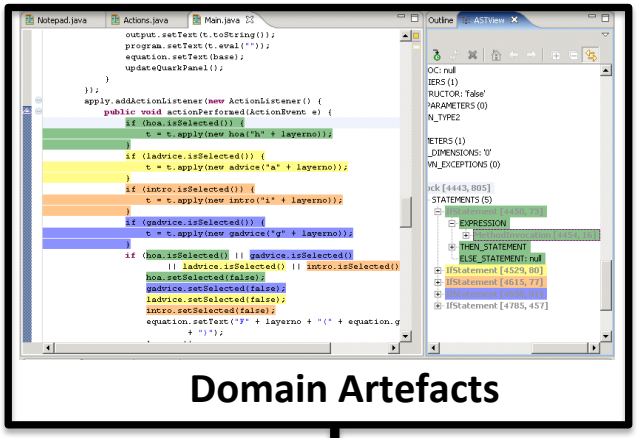
for Software **Mass Customization** [Bass et al., 1998] [Krueger et al., 2001], [Pohl et al., 2005]

Software Product-Line Engineering

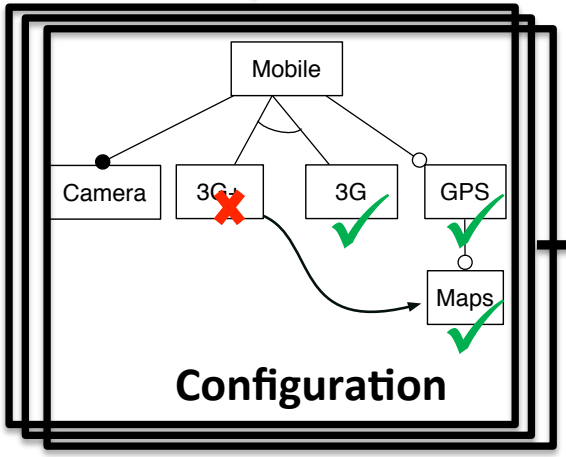


Domain Engineering



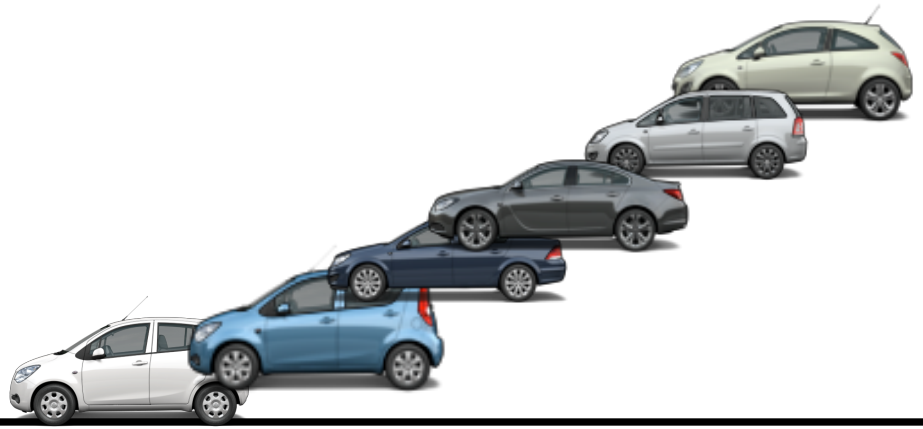


Application Engineering



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

Jan Bosch et al. (2004)



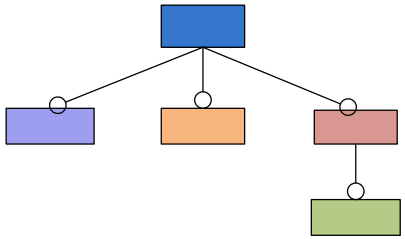
"Reuse-in-the-large works best in families of related systems, and thus is domain dependent." [Glass, 2001]

Domain engineering

Domain Analysis

(problem)

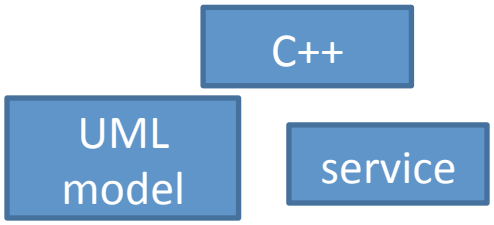
- elicitate requirements and scope the line
- variability modeling: determine commonalities and variabilities usually in terms of features



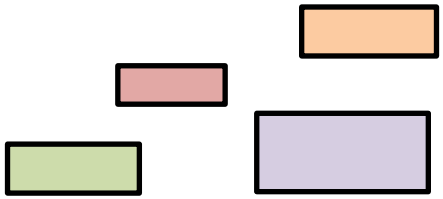
Variability Model (Feature Model)

Domain Implementation

(solution)



Common assets

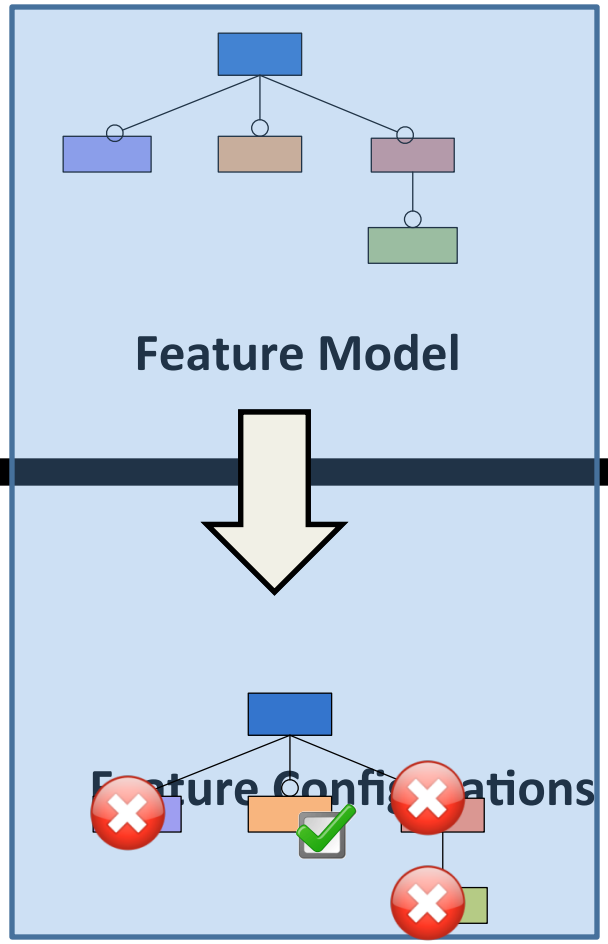


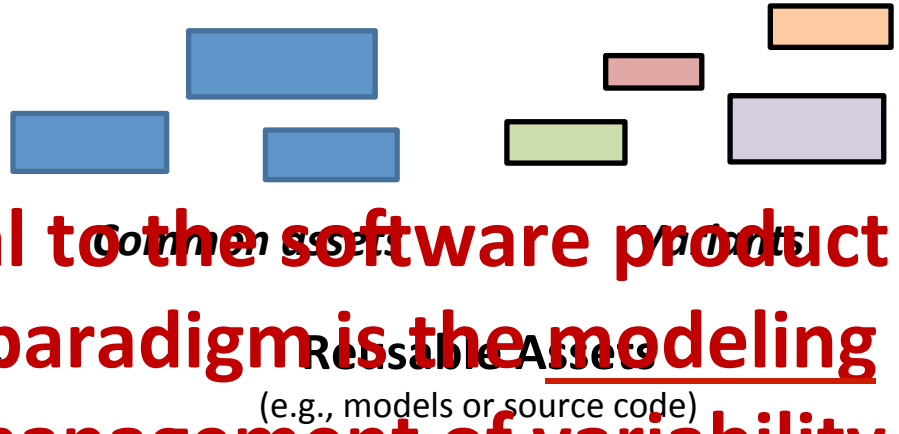
Variants

Reusable Assets

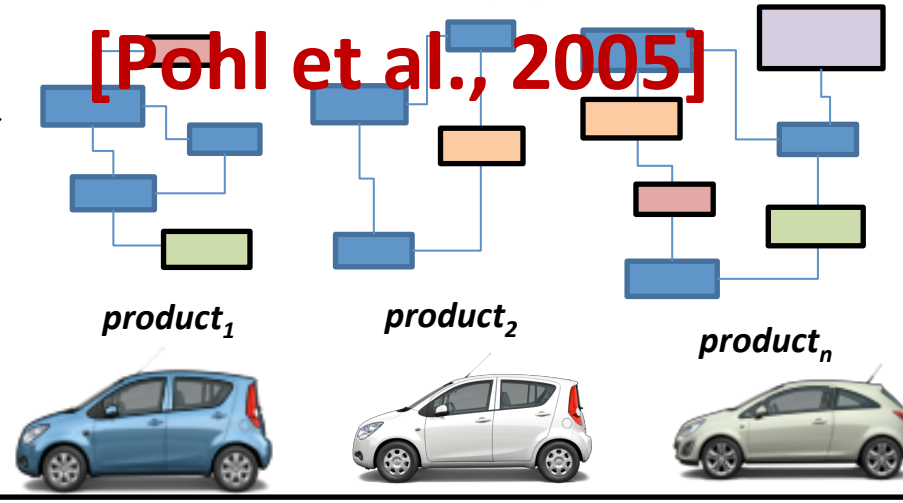
(e.g., models or source code)

Domain engineering (development for reuse)



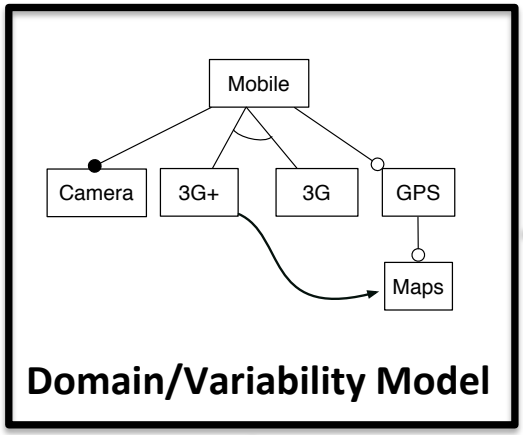


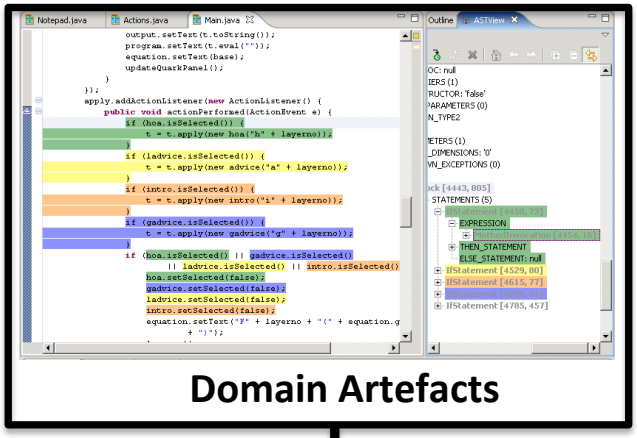
"central to the software product line paradigm is the modeling and management of variability, that is, the commonalities and differences in the applications" [Pohl et al., 2005]



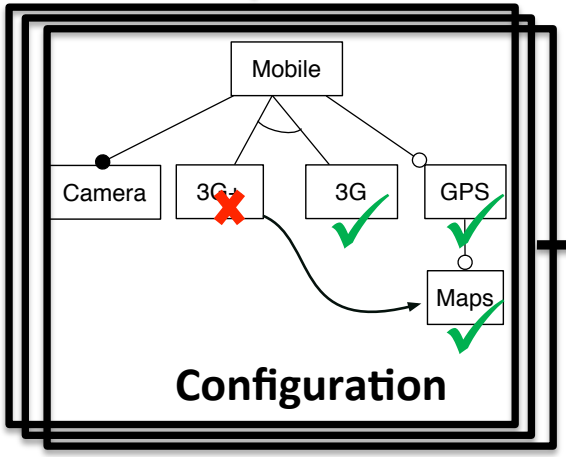
Application engineering (development with reuse)

Domain Engineering





Application Engineering

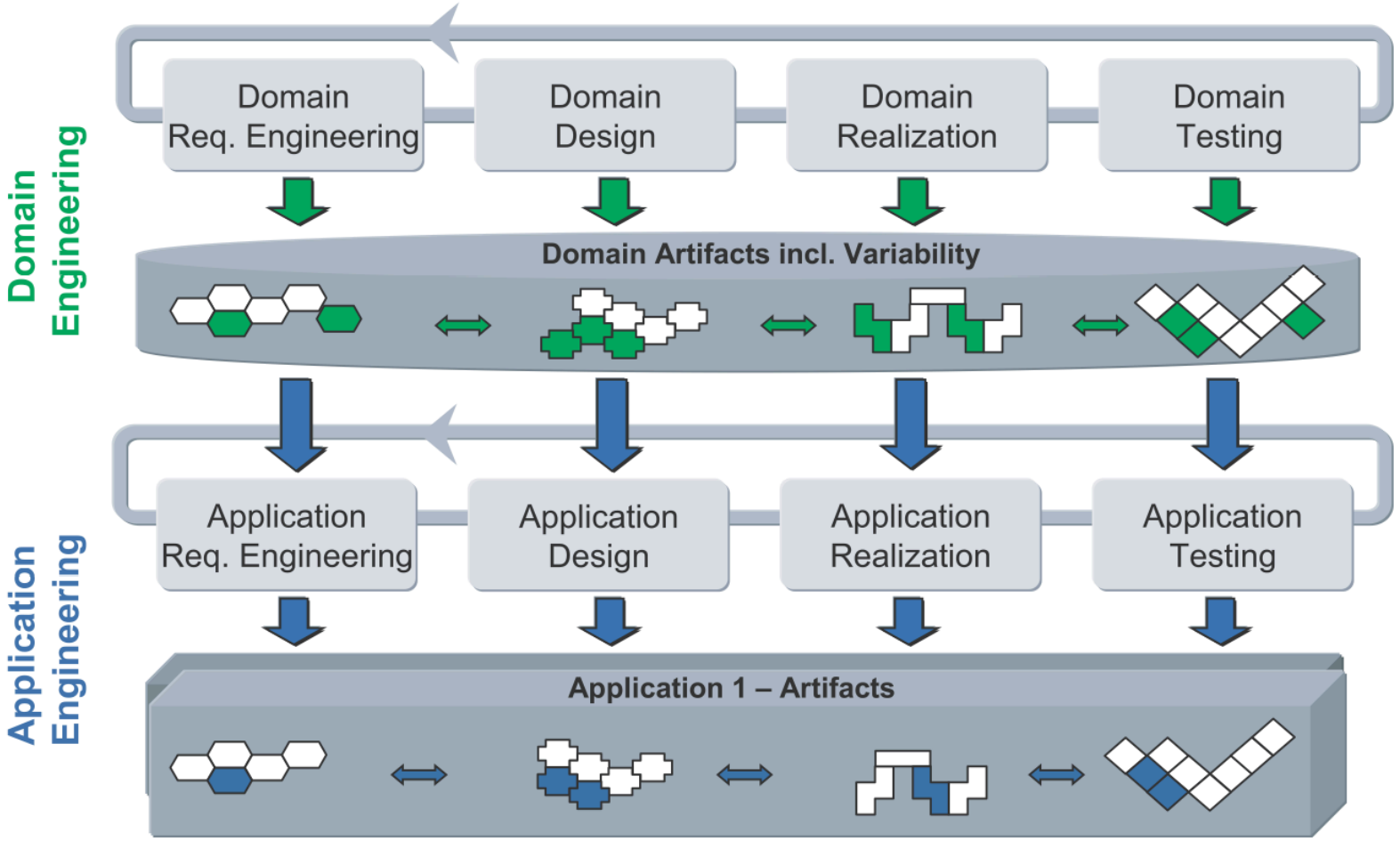


*« the investments required to develop the reusable artifacts during **domain engineering**, are outweighed by the benefits of deriving the individual products during **application engineering** »*

Jan Bosch et al. (2004)

Activities related to domain engineering and application engineering

Software Product-Line Engineering



Domain Analysis

- Collect relevant domain information
  - domain experts (interviews, workshops)
  - system handbooks, textbooks, prototyping, experiments,
  - already known requirements on future systems
  - Creative activity
- Domain Definition
  - examples of systems in a domain,
  - counterexamples (i.e. systems outside the domain),
  - generic rules of inclusion or exclusion (e.g. “Any system having the capability X belongs to the domain.”).
- Domain vocabulary
- Domain concepts
- and integrate it into a coherent *domain model*
  - more or less formal

Czarnecki and Eisenecker (2000)

Domain Modeling (aka Metamodeling)

- Ontology, ER, UML, Ecore, Feature Model

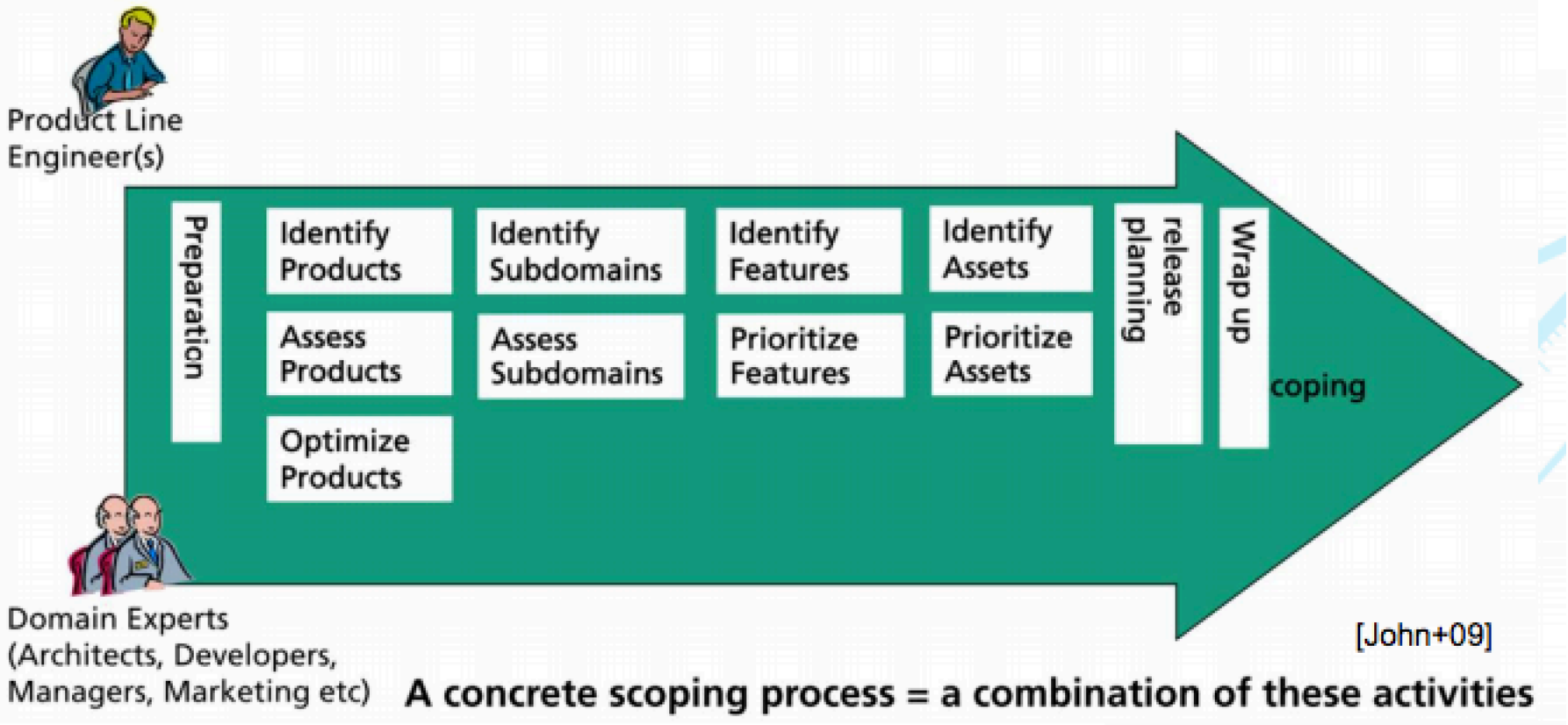
- Analysis of similarity
  - Analyze similarities between entities, activities, events, relationships, structures, etc.
- Analysis of variations
  - Analyze variations between entities, activities, events, relationships, structures, etc.

- Clustering
- Abstraction
- Classification
- Generalization
- Vocabulary construction

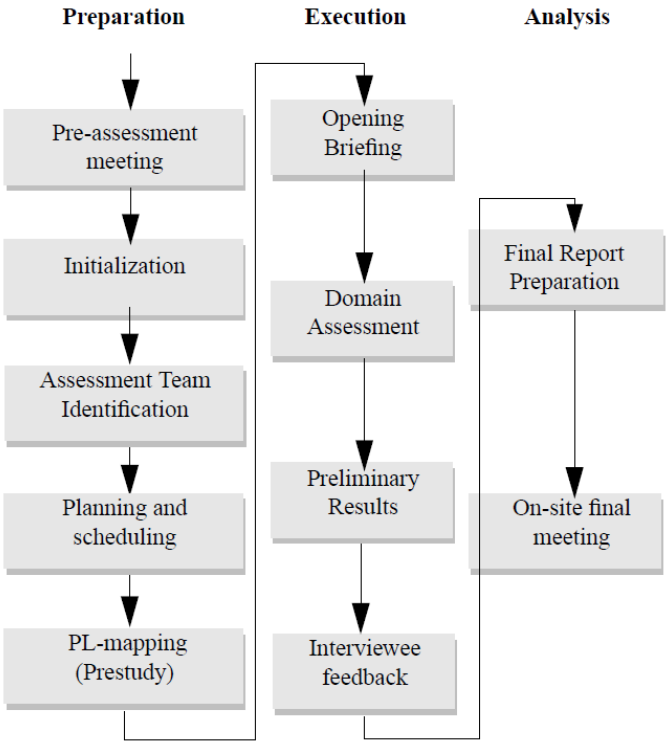
Unused flexibility

Illegal variant

Scoping Activities



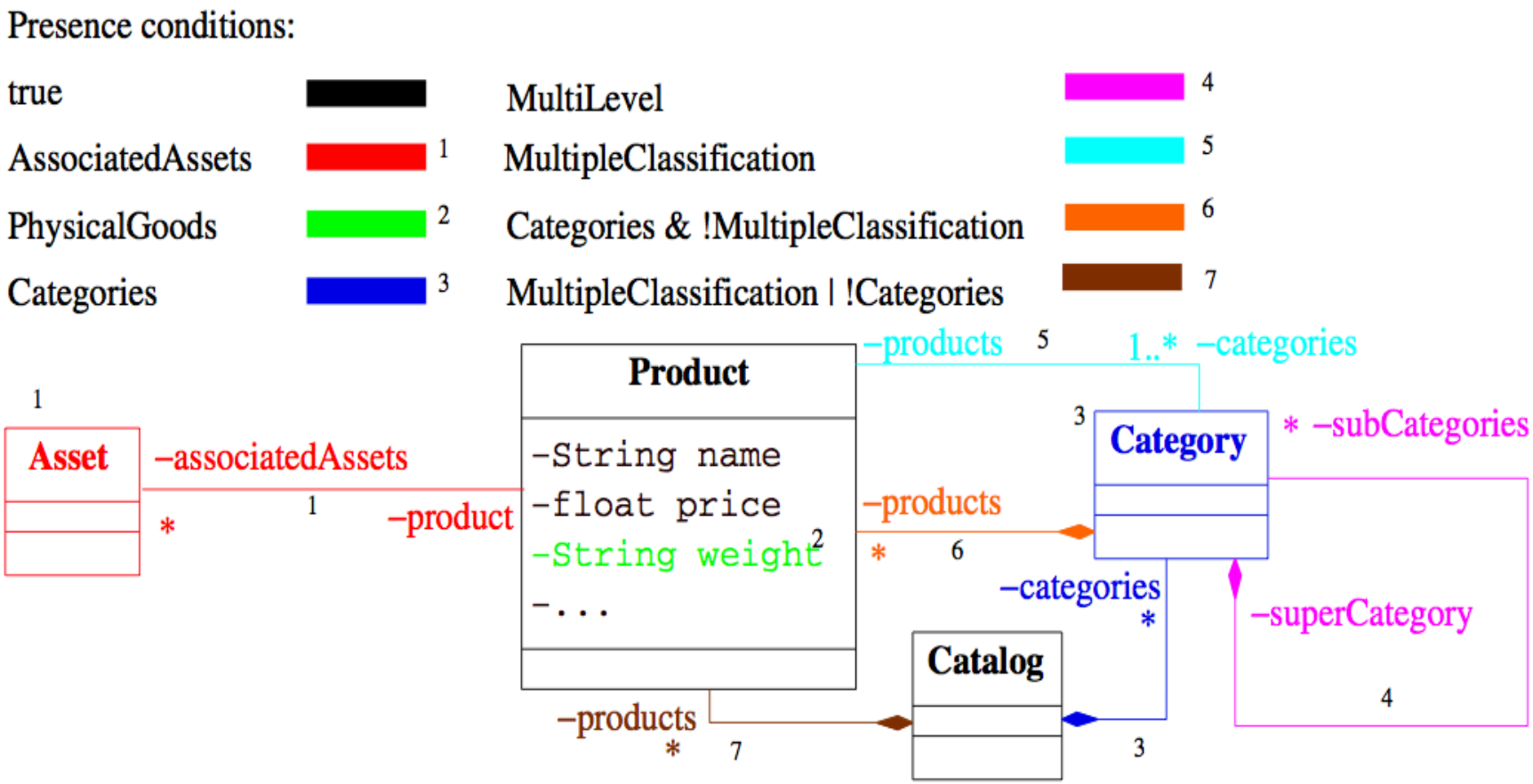
Domain/Product Line Scoping



Schmid 2002

			exist.	planned		potent.
			P1	P2	P3	P4
Domain 1	Sub-Domain 1.1	Feature 1.1.1	X	X	X	X
		Feature 1.1.2	—	X	X	X
		Feature 1.1.3	X	X	—	X
	...	...	...	...	...	...
	Sub-Domain 1.n	Feature 1.n.1	X	—	X	X
		...	...	...	...	...
Domain 2	Sub-Domain 2.1	Feature 2.1.1	—	X	X	—
	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	Feature m.1.1	—	X	—	X

Domain Design

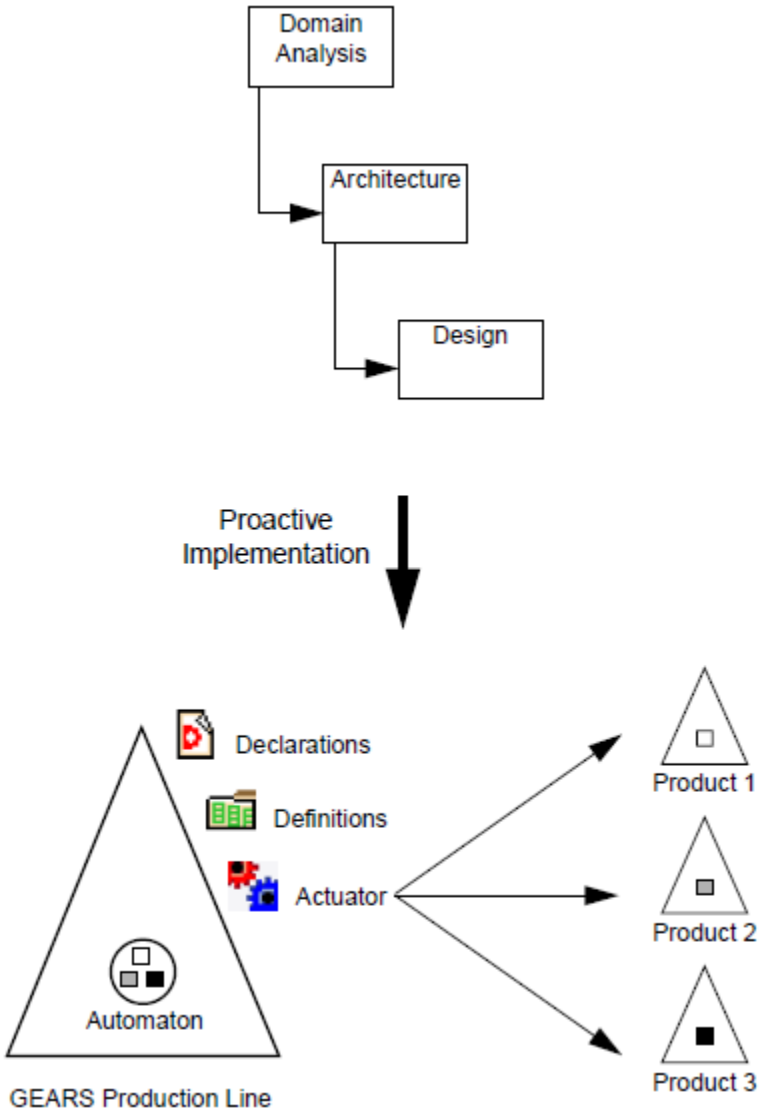


Czarnecki et al. (GPCE'05)

Adoption and Strategies

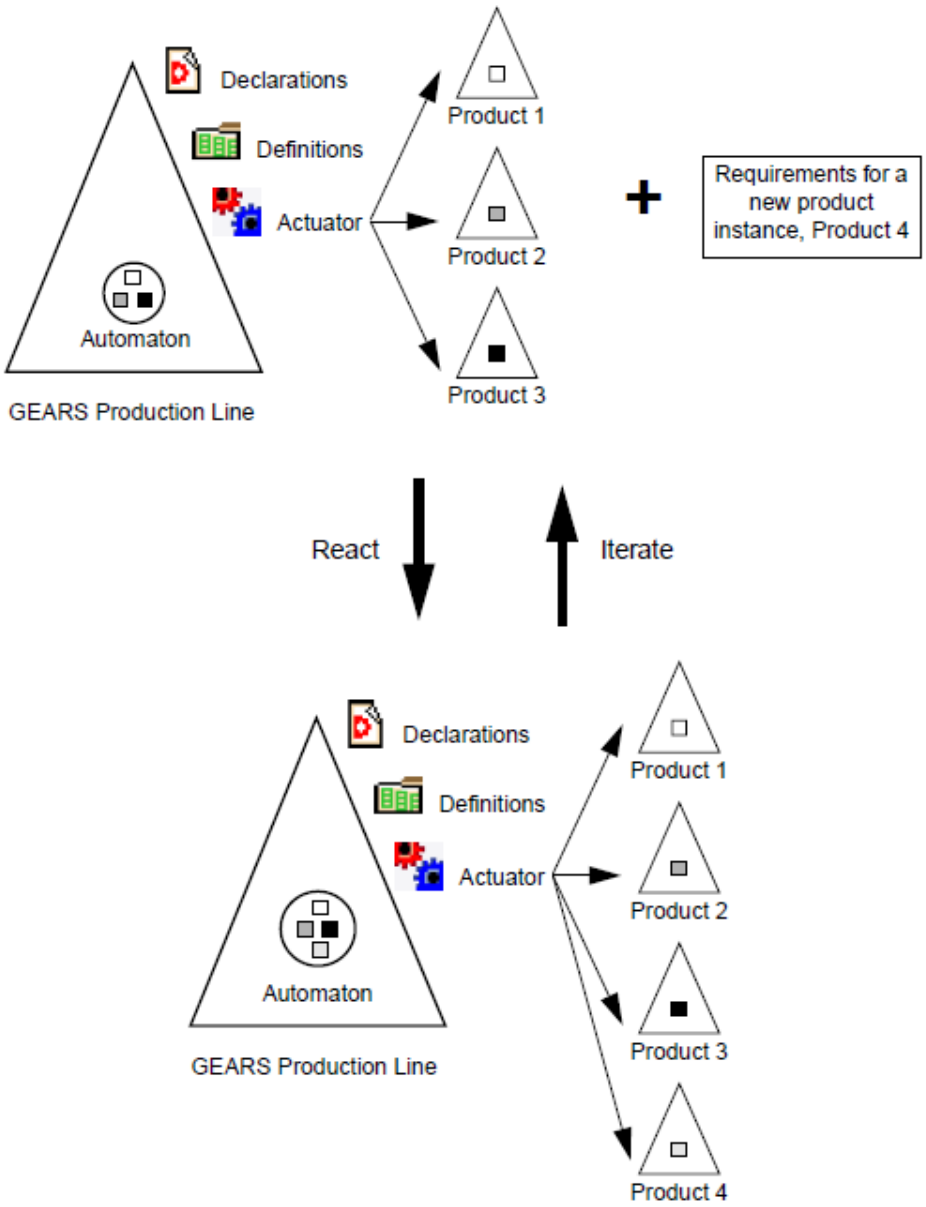
- **Proactive (starting from scratch)**
- **Extractive (re-engineering, from products to product line)**
- **Reactive (hybrid)**

Proactive



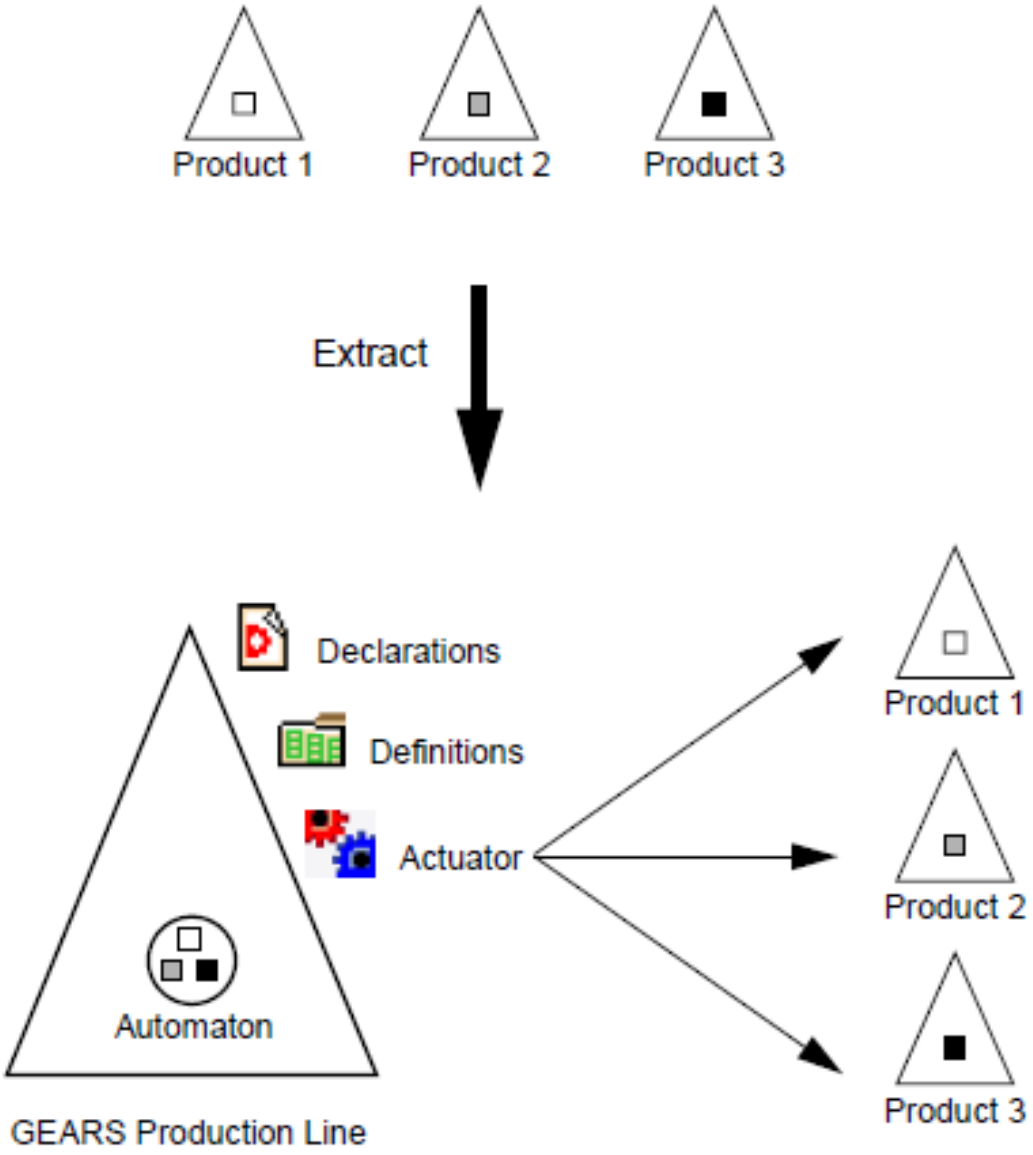
[Krueger 2002]

Reactive

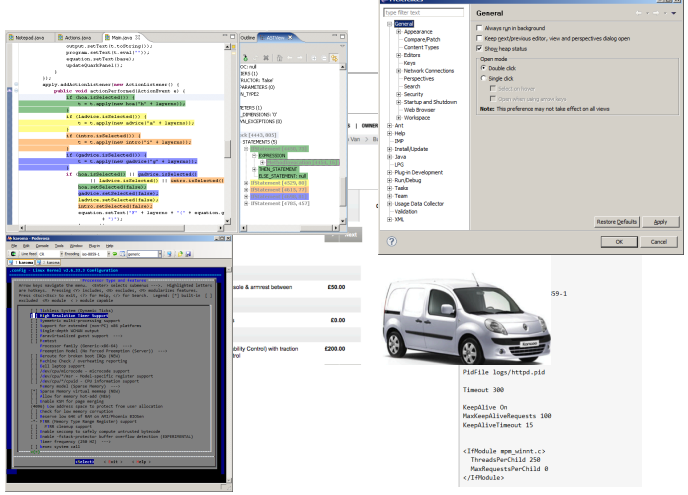


[Krueger 2002]

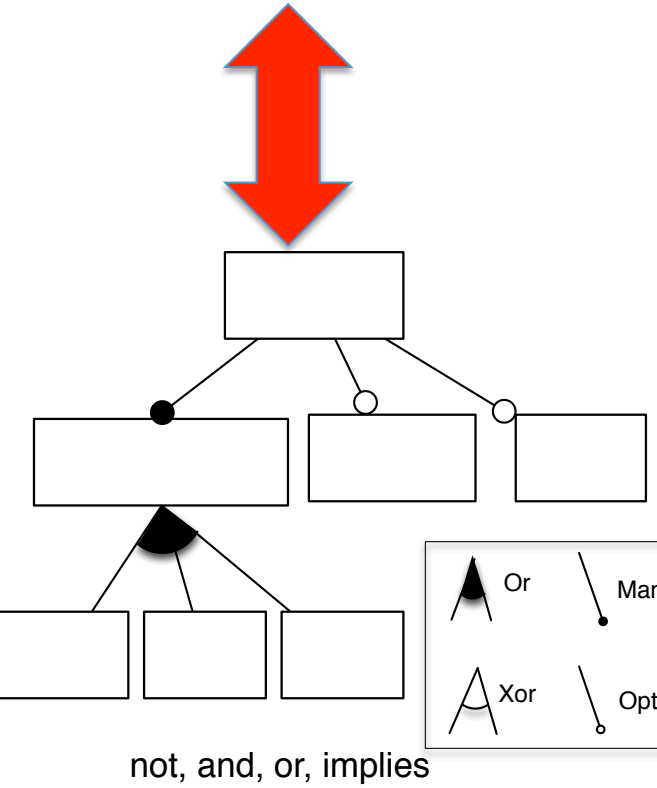
Extractive



[Krueger 2002]

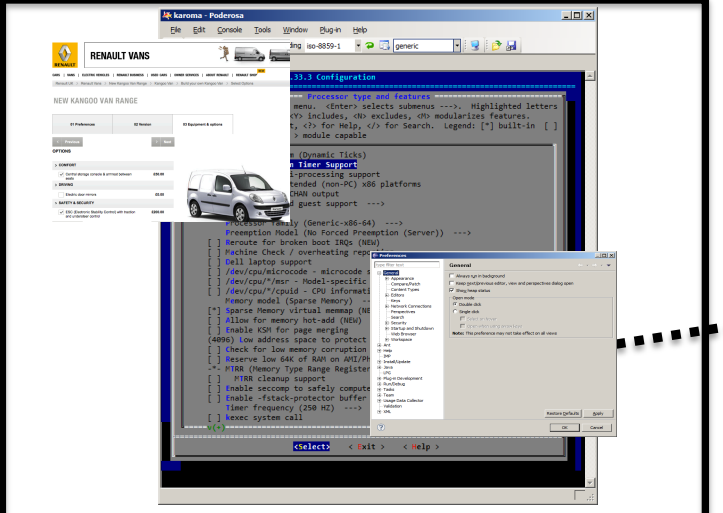


Variants of code (e.g., Java ou C)
Variants of user interfaces
Variants of video sequences
Variants of models (e.g., UML or SysML)
…



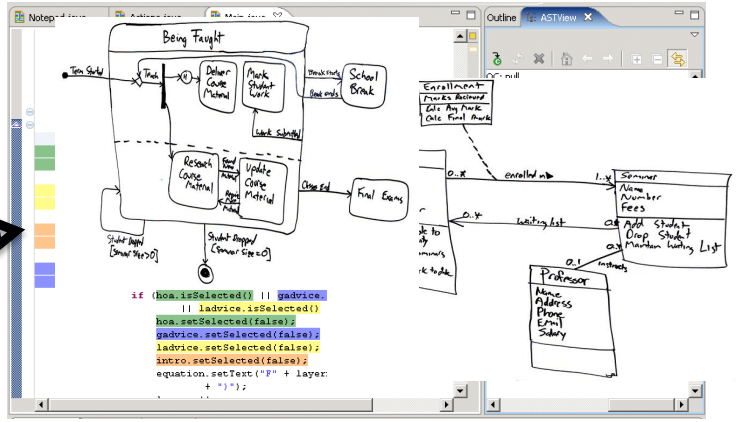
not, and, or, implies

Variability Models (feature models)

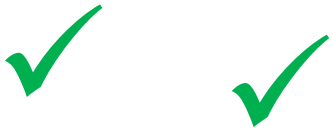


Variability Model

mapping



Base Artefacts (e.g., models)



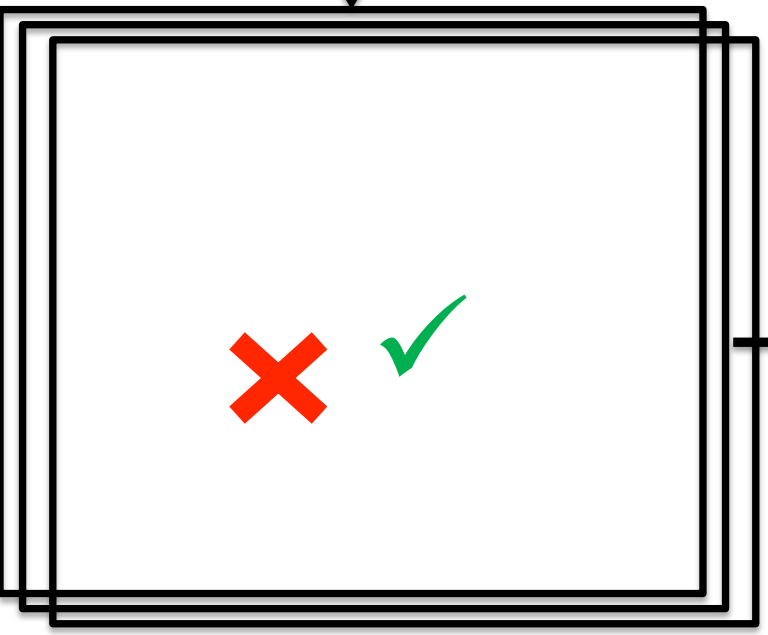
Configuration

Software Generator
(derivation engine)

Variability Model

mapping

Base Artefacts



Software Generator
(derivation engine)

Branch: **master**

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

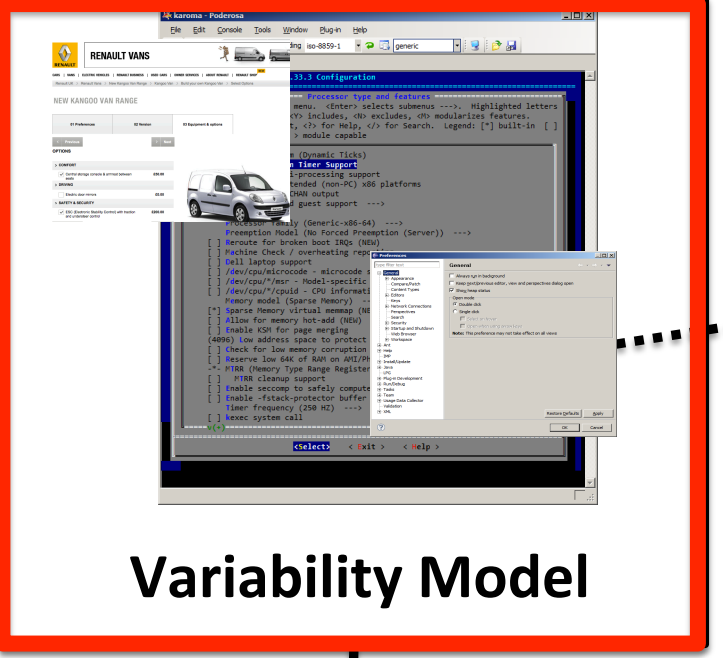
jdubois 2 days ago Use Spring Boot's configuration meta-data

9 contributors

184 lines (165 sloc) | 9.69 KB

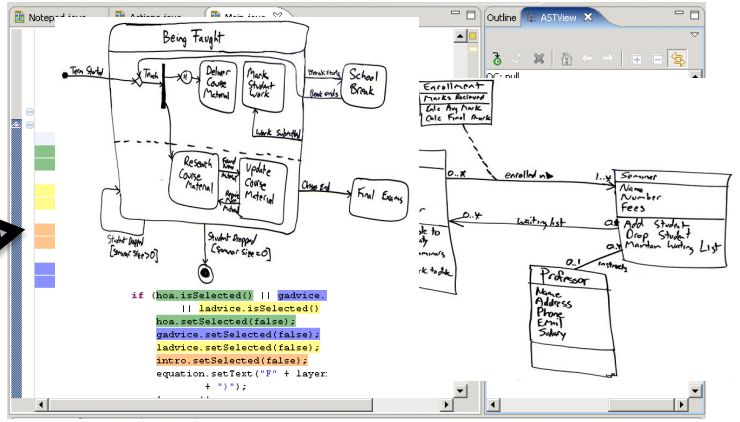
Raw | Blame | History

```
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') { %>
```

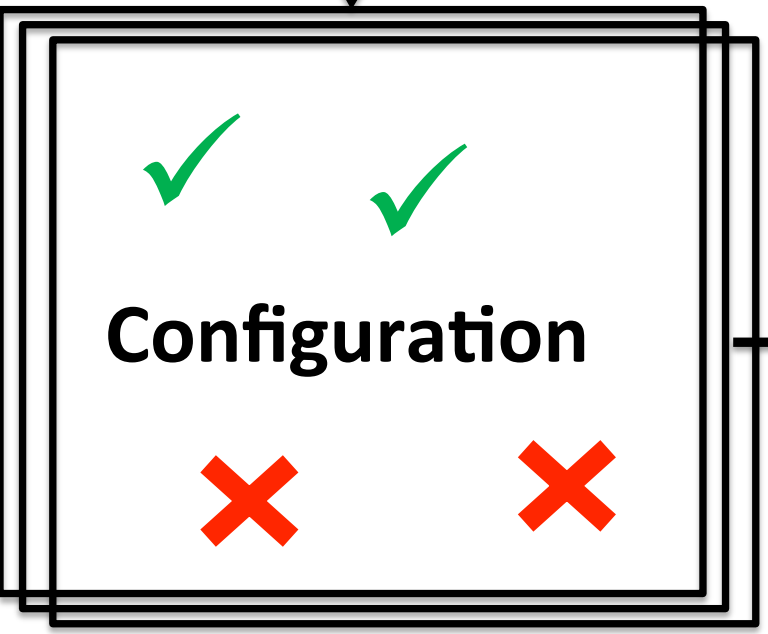


Variability Model

mapping



Base Artefacts (e.g., models)



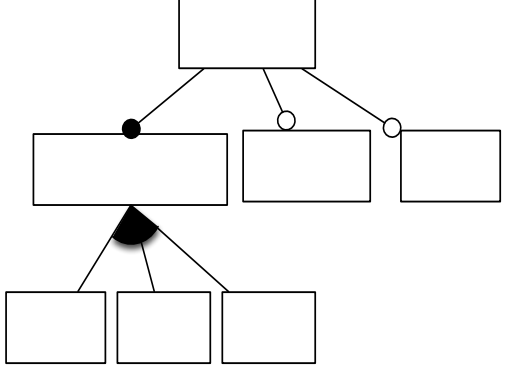
Configuration

Software Generator
(derivation engine)

Unused flexibility

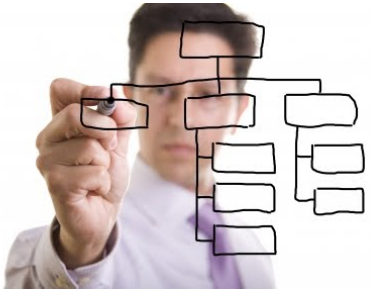
Illegal variant

Feature Model

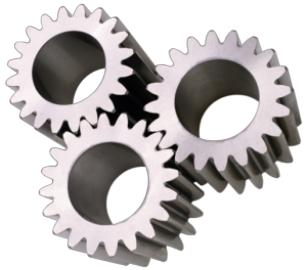


not, and, or, implies

Communicative



Analytic



Generative

Homework

- Jhipster (http://www.jhipster.tech/), let us consider the last version (September 6, 2017)

1. List all options of Jhipster
2. Propose some options that Jhipster could propose/ support
3. Describe some combinations of options of Jhipster that are not possible; justify why!

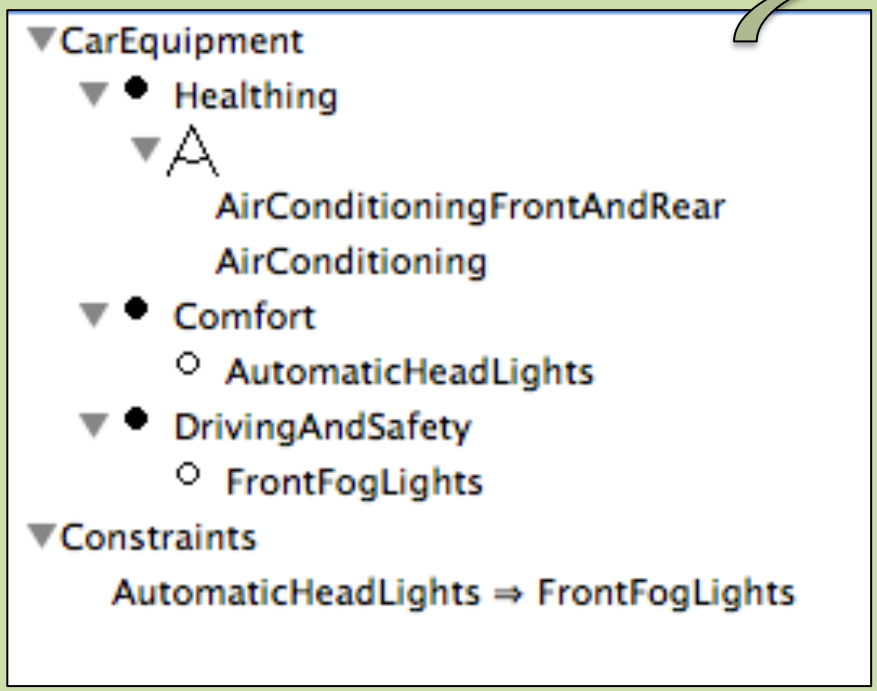
Look at homepage, documentation, and… source code!

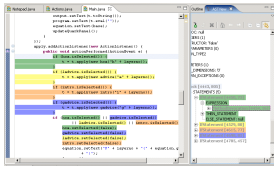
- Upload your contributions (document) here http://tinyurl.com/jhipster-MIAGE-RE1718

(deadline: next week)

Feature Models

(defacto standard for modeling variability)

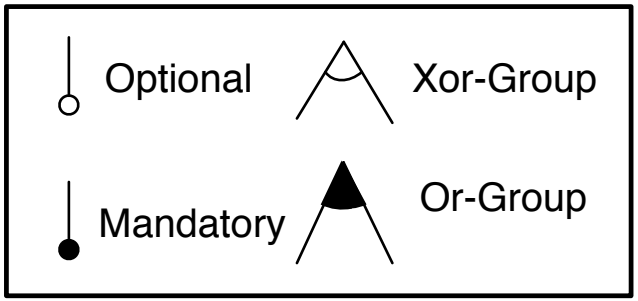




Hierarchy: rooted tree

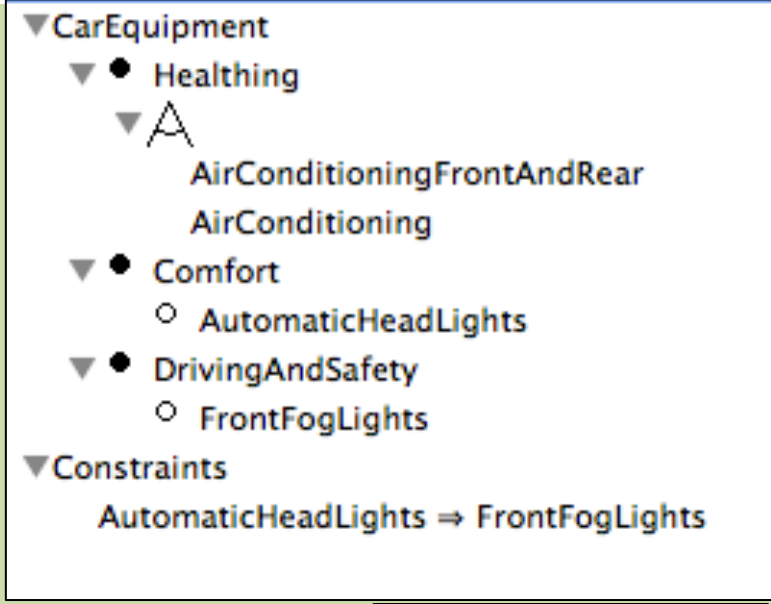
Variability:

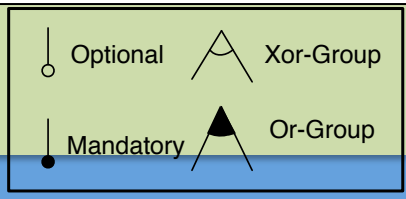
- mandatory,
- optional,
- Groups: exclusive or inclusive features
- Cross-tree constraints



SPLC12Scripts Model | SPLC12Scripts.config

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





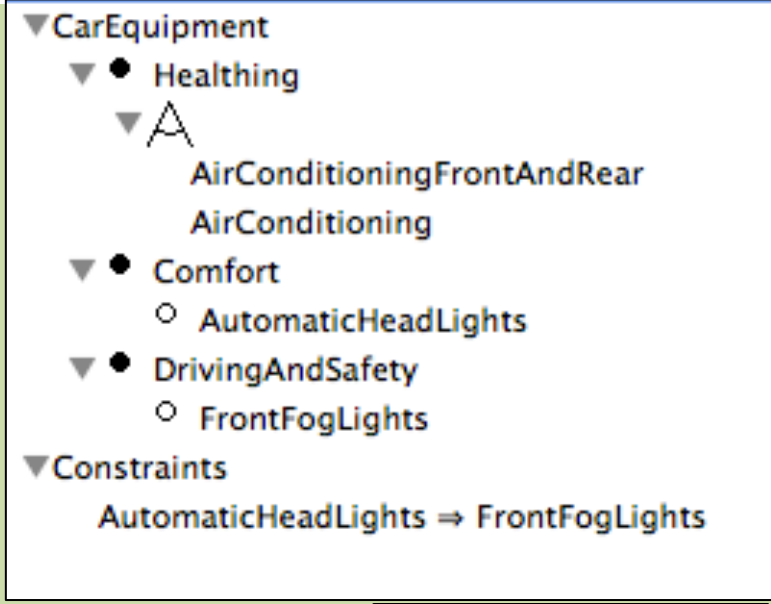
Hierarchy + Variability
=
set of valid configurations

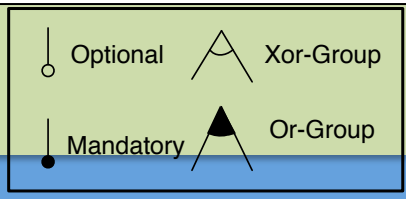
configuration = set of features selected

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

SPLC12Scripts Model | SPLC12Scripts.config

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

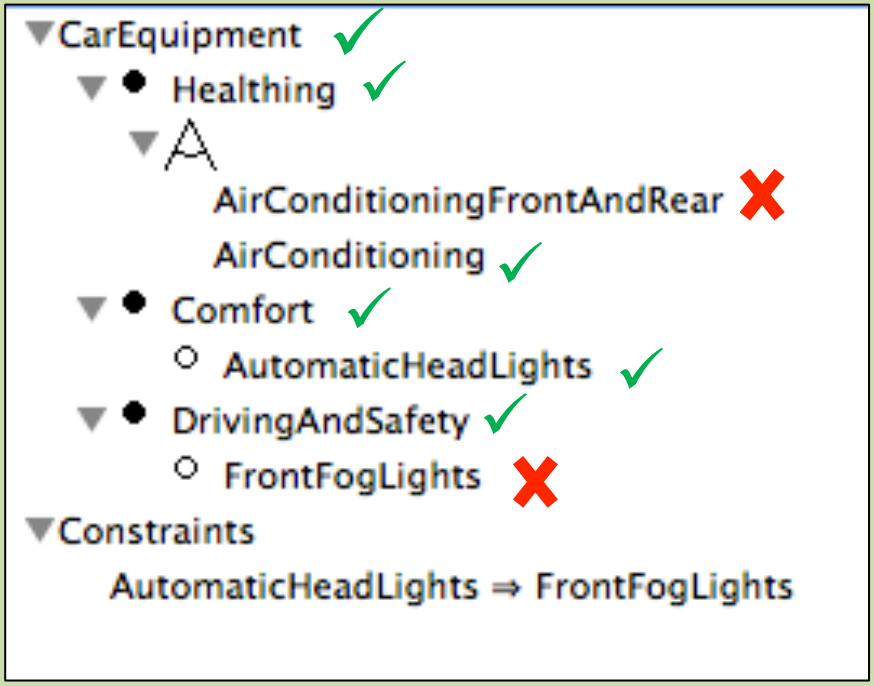


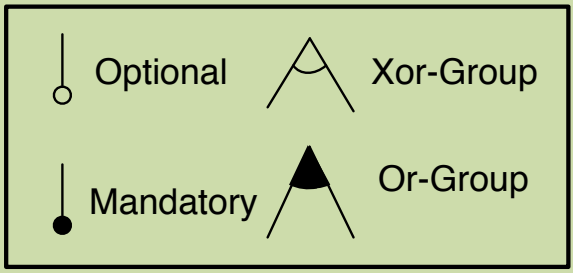


Hierarchy + Variability
=
set of valid configurations

configuration = set of features selected

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning}

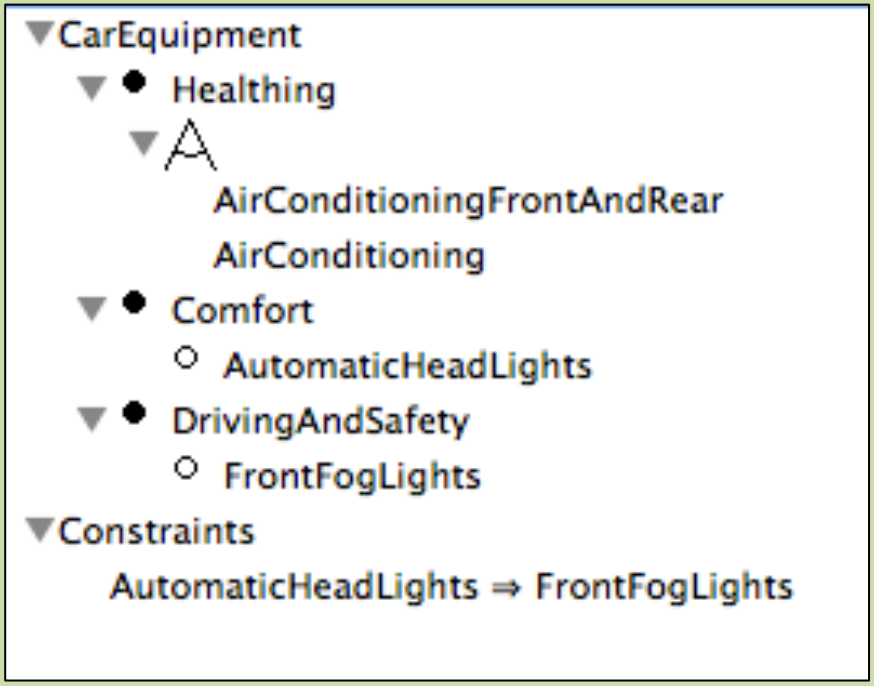


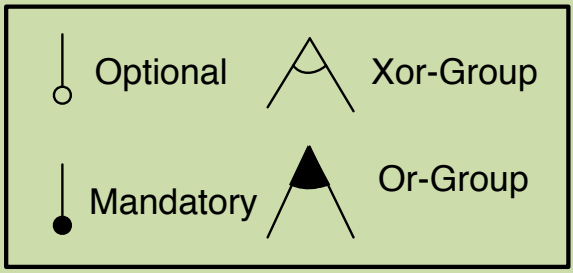


Hierarchy + Variability
=
set of valid configurations

configuration = set of features selected

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



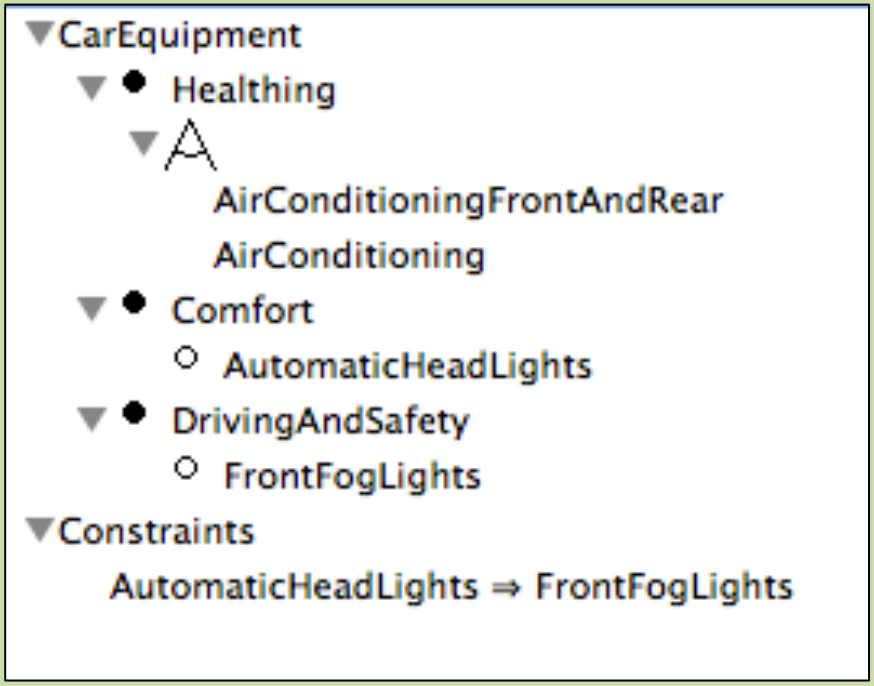


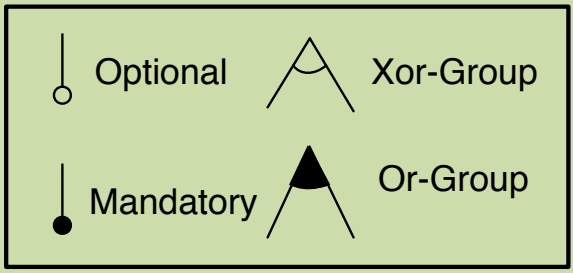
Hierarchy + Variability
=
set of valid configurations

{CarEquipment, Comfort, DrivingAndSafety, Healthing}

X

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

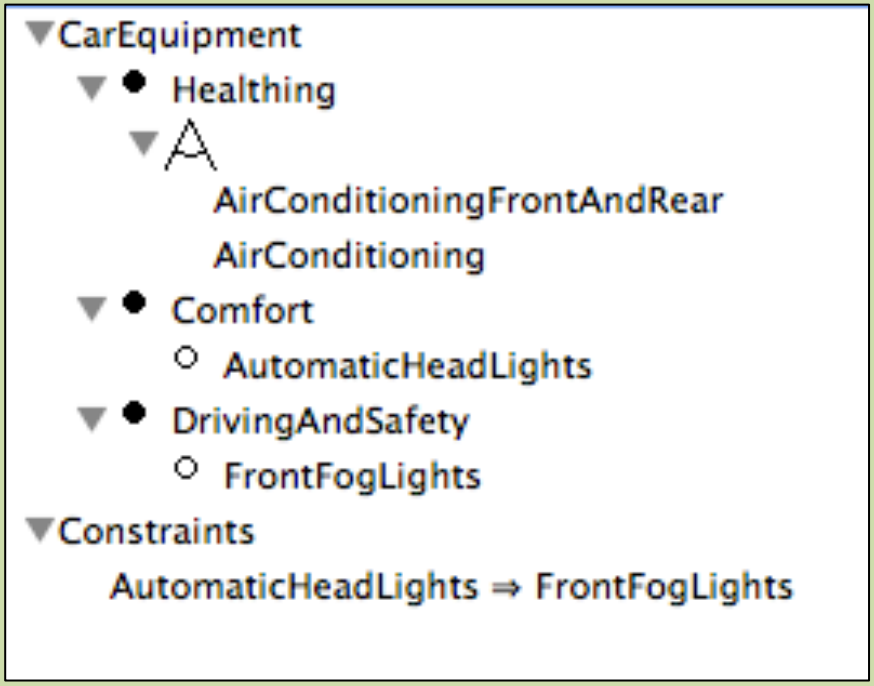


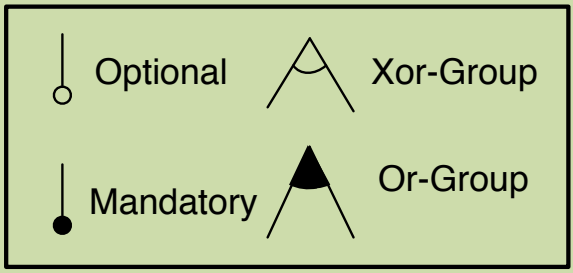


Hierarchy + Variability
=
set of valid configurations

Configuration set (from a basic feature model of car)

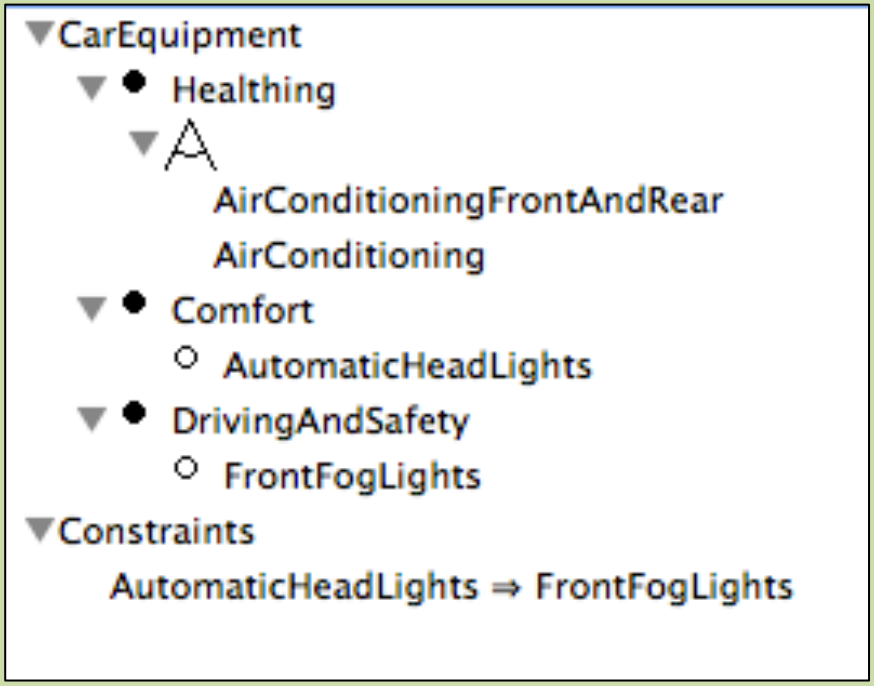
	CarEquipment	Comfort	DrivingAndSafety	Healting	AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Car2	yes	yes	yes	yes	yes	yes	yes	no
Car6	yes	yes	yes	yes	no	yes	no	yes
Car1	yes	yes	yes	yes	yes	yes	no	no
Car4	yes	yes	yes	yes	no	no	no	yes
Car5	yes	yes	yes	yes	yes	no	no	no
Car3	yes	yes	yes	yes	no	yes	yes	yes

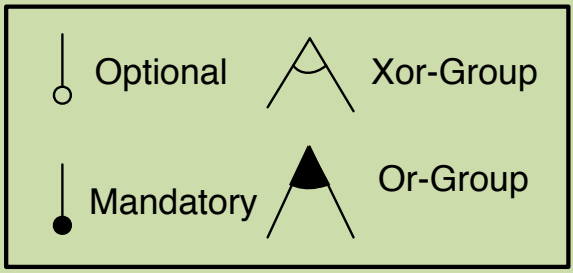




Hierarchy + Variability
=
set of valid configurations

Product	CarEquipment	Comfort	DrivingAndSafety	Healting	AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Find	Yes / No	Yes / No	Yes / No	Yes / No	Yes / No	Yes / No	Yes / No	Yes / No
Car1	yes	yes	yes	yes	yes	yes	no	no
Car2	yes	yes	yes	yes	yes	yes	yes	no
Car3	yes	yes	yes	yes	no	yes	yes	yes
Car4	yes	yes	yes	yes	no	no	no	yes
Car5	yes	yes	yes	yes	yes	no	no	no
Car6	yes	yes	yes	yes	no	yes	no	yes



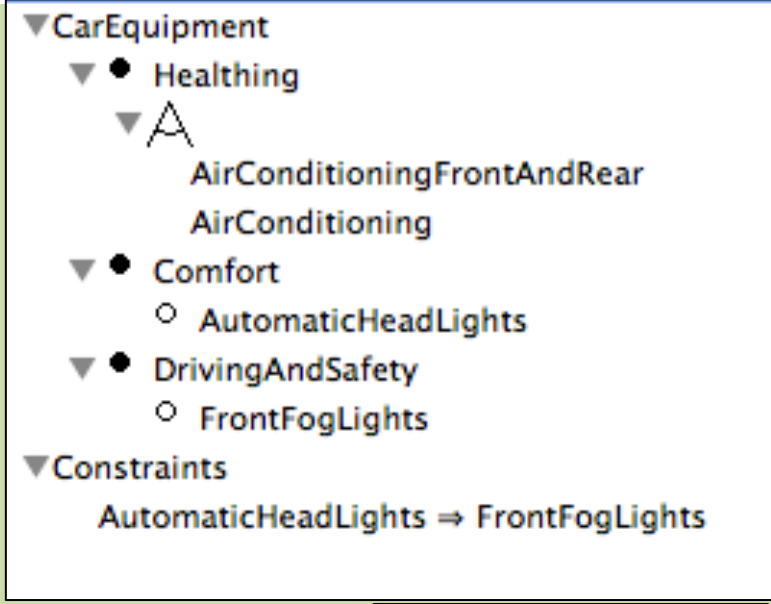


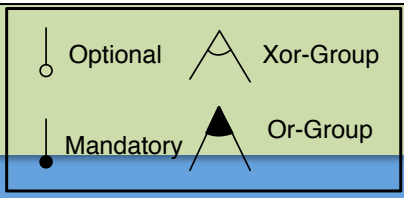
Hierarchy + Variability
=
set of valid configurations

Product					AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Find					Yes No	Yes No	Yes No	Yes No
Car1					yes	yes	no	no
Car2					yes	yes	yes	no
Car3					no	yes	yes	yes
Car4					no	no	no	yes
Car5					yes	no	no	no
Car6					no	yes	no	yes

SPLC12Scripts Model | SPLC12Scripts.config

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



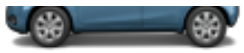


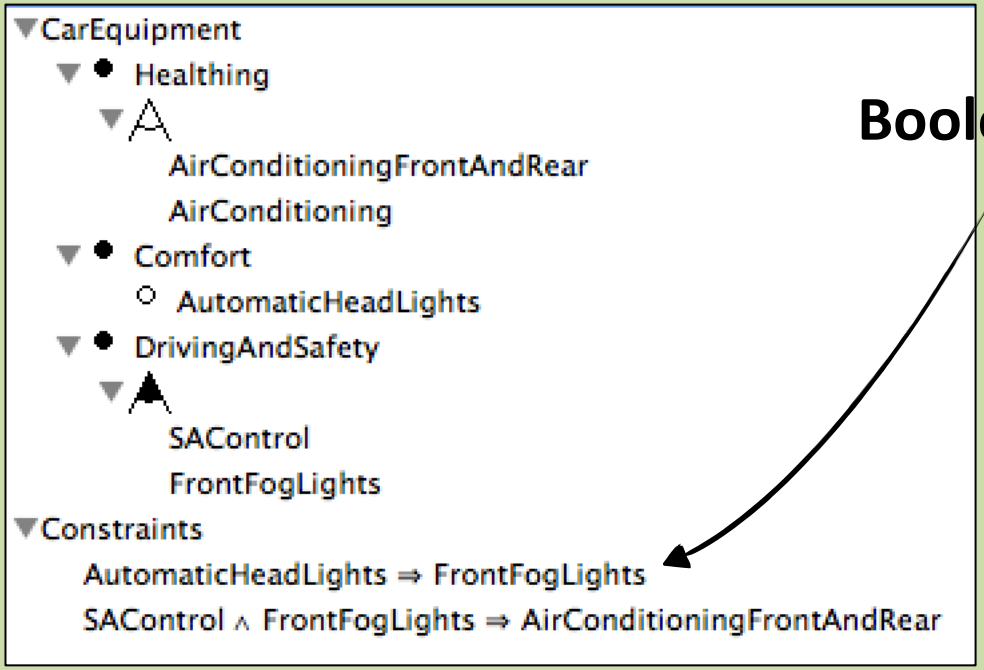
Hierarchy + Variability
=
set of valid configurations

configuration = set of features selected

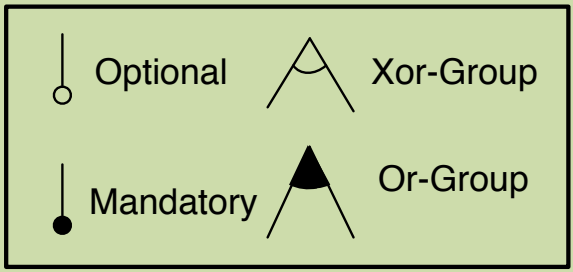
{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning}

Product	AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Find	Yes ☑ No ☐	Yes ☐ No ☑	Yes ☐ No ☑	Yes ☐ No ☑
Car5	yes	no	no	no





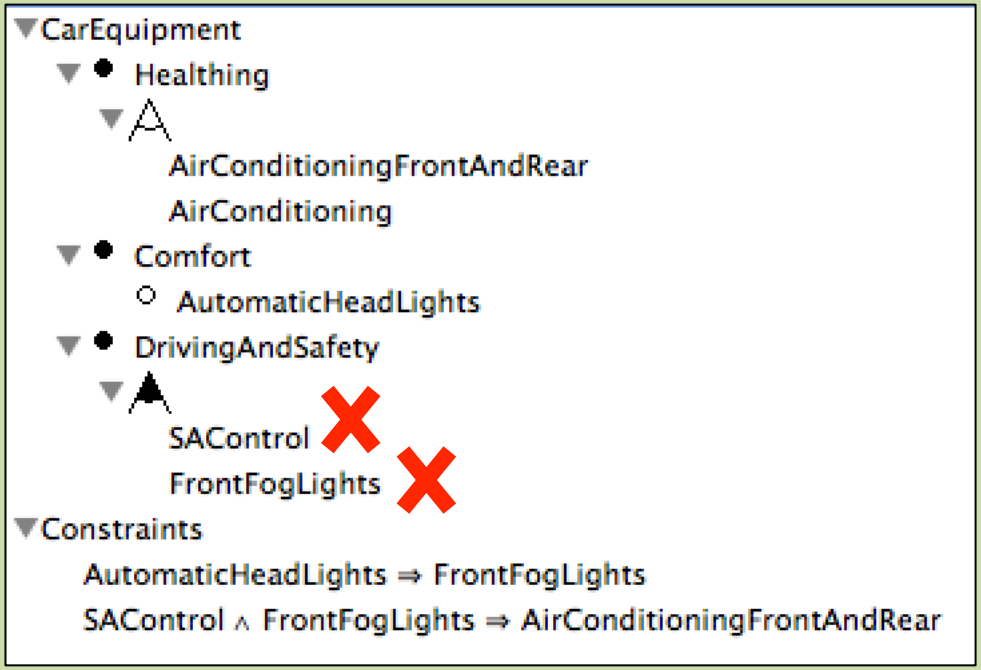
Boolean logic: ^, v, not, implies

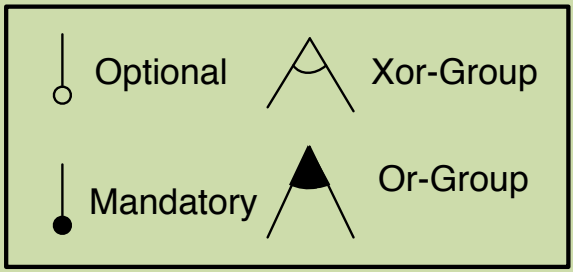


Hierarchy + Variability

=

set of valid configurations



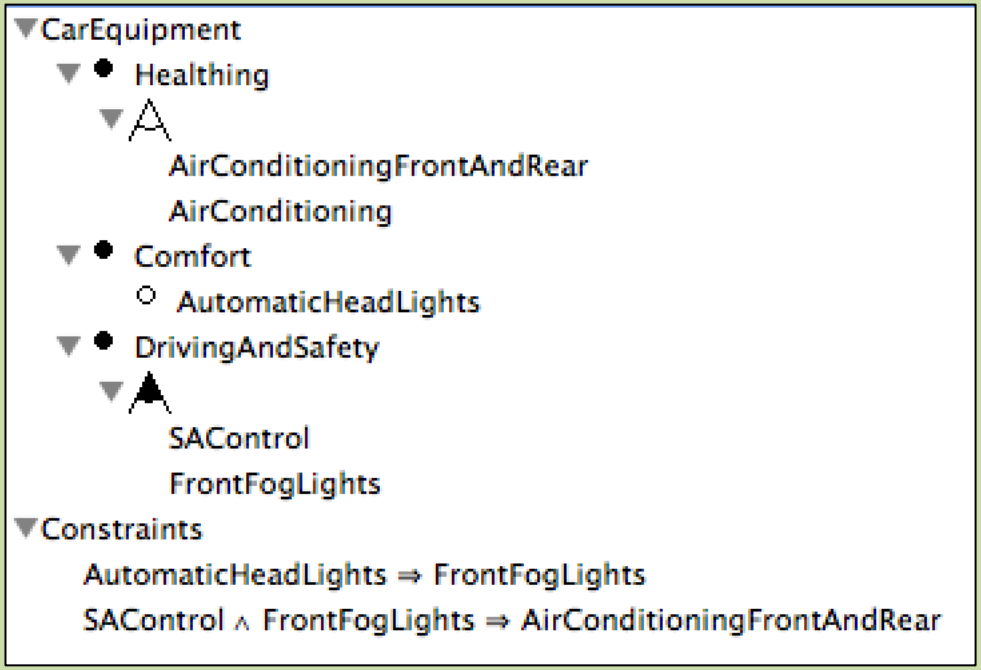


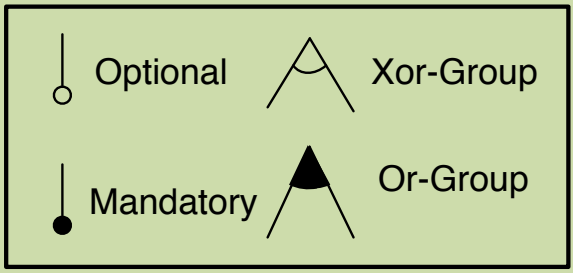
Hierarchy + Variability

=

set of valid configurations

Or-group: at least one!



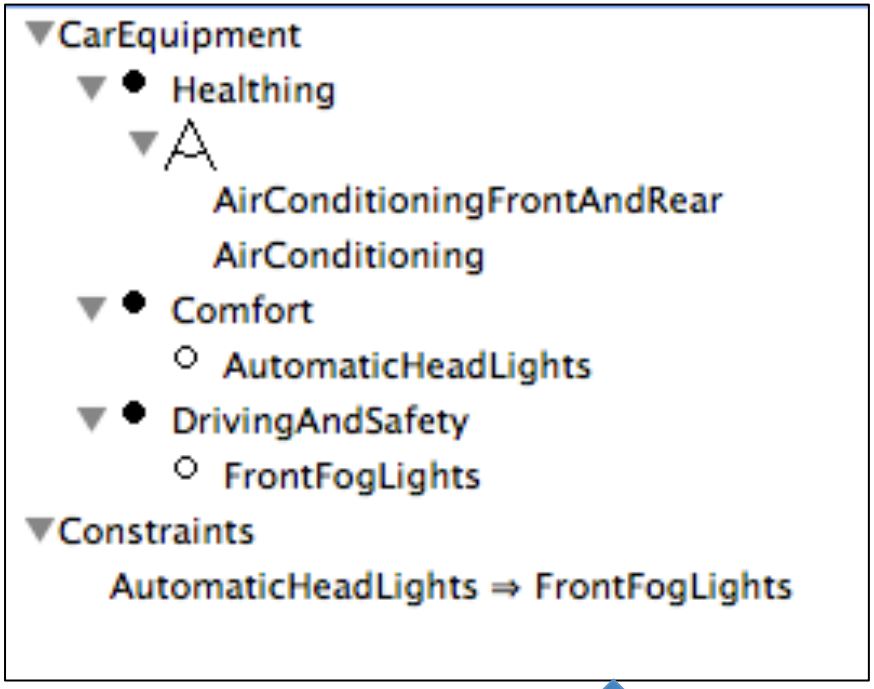


Hierarchy + Variability
=
set of valid configurations

{CarEquipment, Comfort, DrivingAndSafety, Healthing}

X

{AirConditioningFrontAndRear, FrontFogLights, SAControl}
{AirConditioningFrontAndRear, SAControl}
{AutomaticHeadLights, AirConditioning, FrontFogLights}
{AirConditioningFrontAndRear, SAControl, AutomaticHeadLights, FrontFogLights}
{FrontFogLights, AirConditioning}
{AutomaticHeadLights, AirConditioningFrontAndRear, FrontFogLights}
{FrontFogLights, AirConditioningFrontAndRear}
{SAControl, AirConditioning}



(Boolean) Feature Models

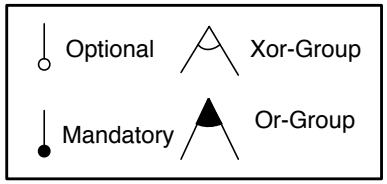
(Boolean) Formula φ

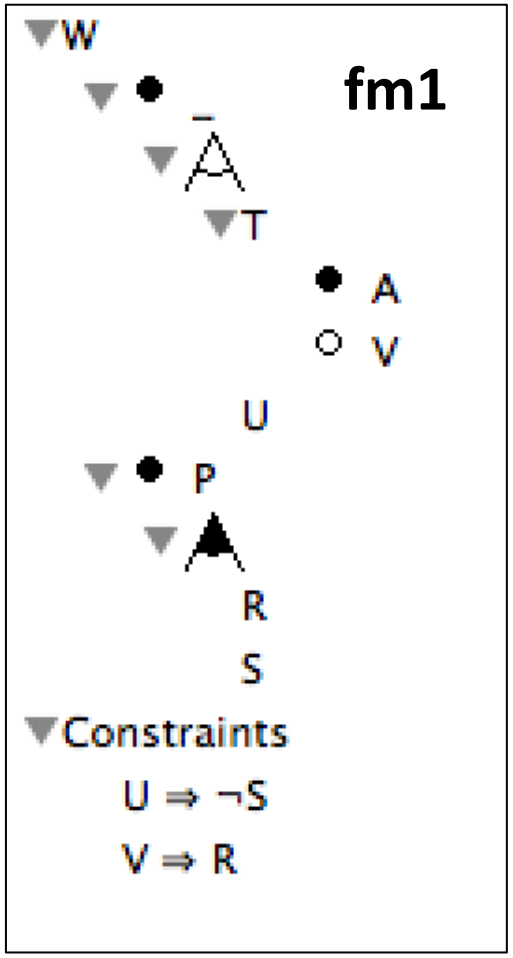
Product	CarEquipment	Comfort	DrivingAndSafety	Healting	AirConditioning	FrontFogLights	AutomaticHeadLights	AirConditioningFrontAndRear
Find	Yes / No	Yes / No	Yes / No	Yes / No	Yes / No	Yes / No	Yes / No	Yes / No
Car1	yes	yes	yes	yes	yes	yes	no	no
Car2	yes	yes	yes	yes	yes	yes	yes	no
Car3	yes	yes	yes	yes	no	yes	yes	yes
Car4	yes	yes	yes	yes	no	no	no	yes
Car5	yes	yes	yes	yes	yes	no	no	no
Car6	yes	yes	yes	yes	no	yes	no	yes

(Boolean) Product Comparison Matrix

(Boolean) Feature Models

Hierarchy + Variability = set of valid configurations

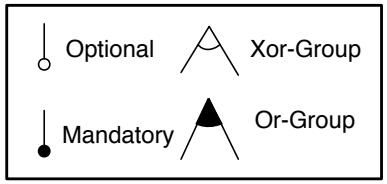


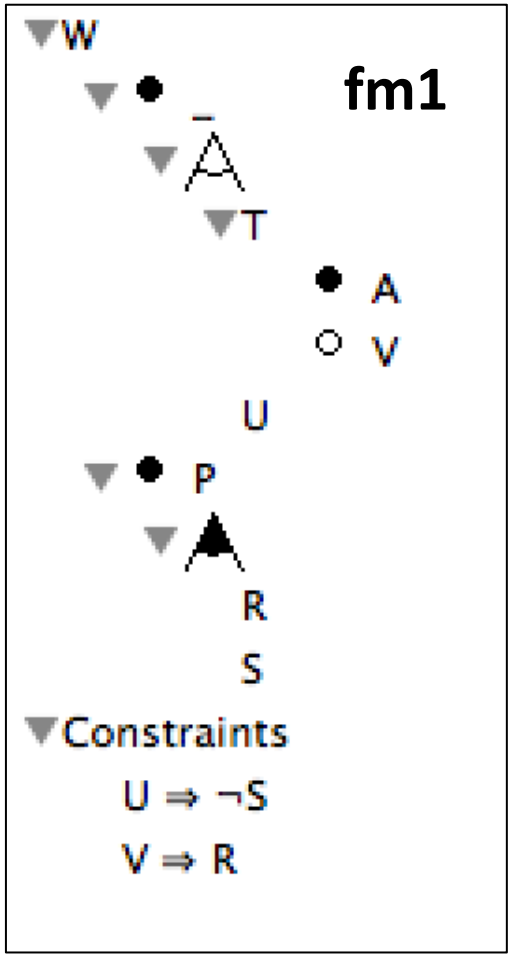


$$[\![fm1]\!] = \{$$
$$\{W, P, R, S, T, A, V\},$$
$$\{W, P, S, T, A\},$$
$$\{W, P, R, T, A\},$$
$$\{W, P, R, U\},$$
$$\{W, P, R, T, V, A\},$$
$$\{W, P, R, S, T, A\},$$
$$\}$$

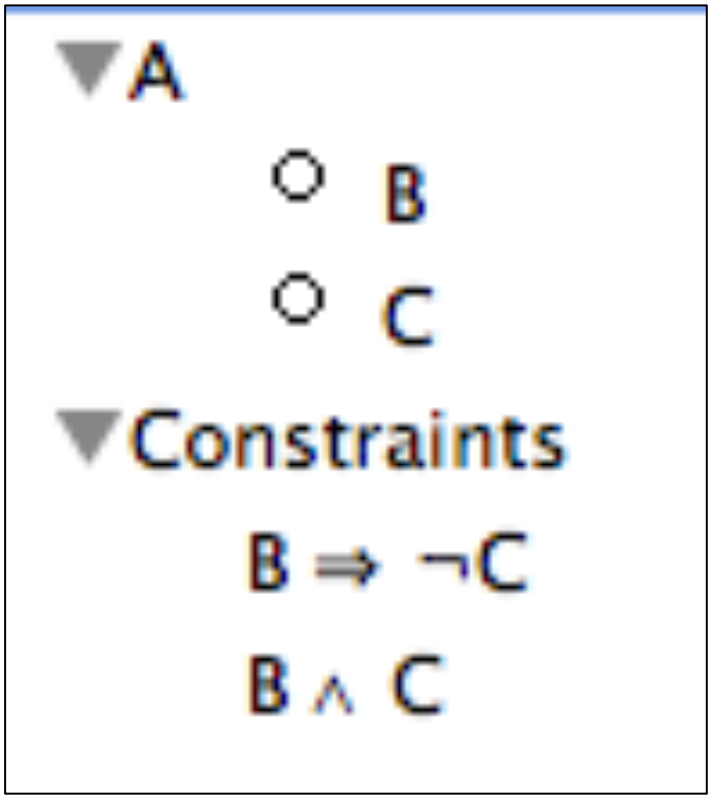
(Boolean) Feature Models

~ Boolean formula

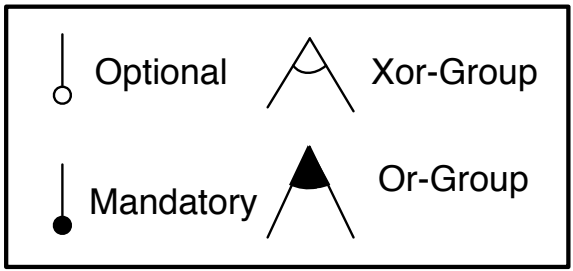


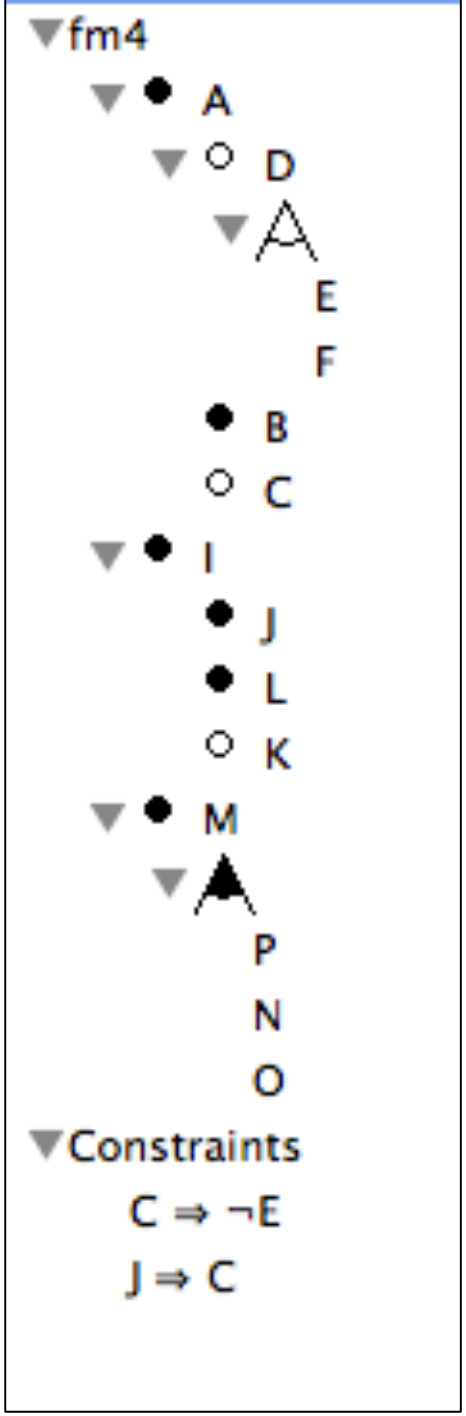


$$
\begin{aligned}
&\phi_{fm_1} = W \;//\; \text{root} \\
&\wedge\, W \Leftrightarrow P \;//\; \text{mandatory} \\
&//\; \text{Or-group} \\
&\wedge\, P \Rightarrow R \vee S \\
&\wedge\, R \Rightarrow P \wedge\; S \Rightarrow P \\
&\wedge\, V \Rightarrow T \;//\; \text{optional} \\
&\wedge\, A \Leftrightarrow T \;//\; \text{mandatory} \\
&//\; \text{Xor-group} \\
&\wedge\, T \Rightarrow W \\
&\wedge\, U \Rightarrow W \\
&\wedge\, \neg T \vee \neg U \\
&//\; \text{constraints} \\
&\wedge\, V \Rightarrow R \;//\; \text{implies} \\
&\wedge\, \neg U \Rightarrow \neg S \;//\; \text{excludes}
\end{aligned}
$$



Empty set of configurations





Dead feature

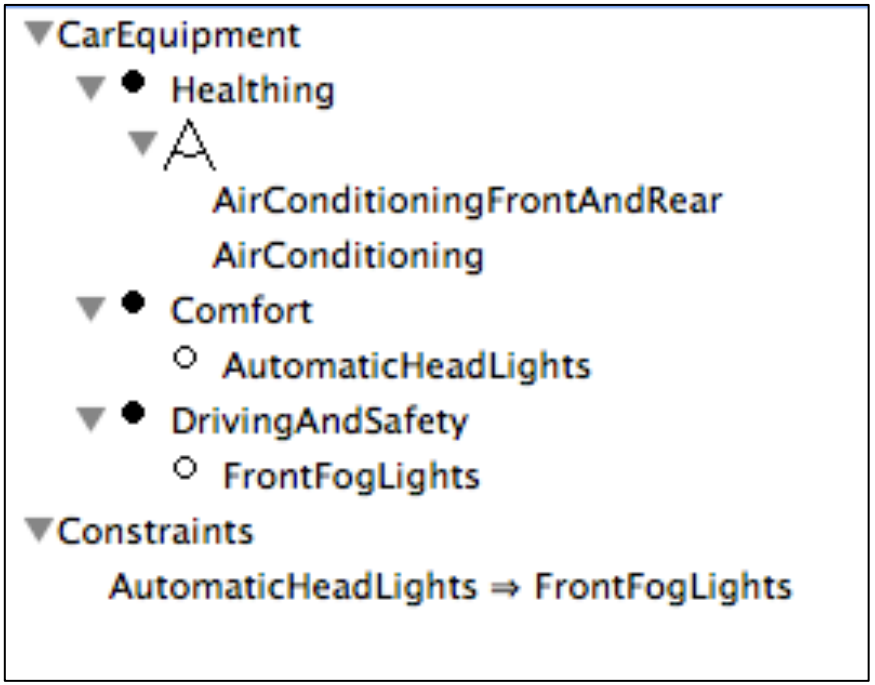
False optional feature

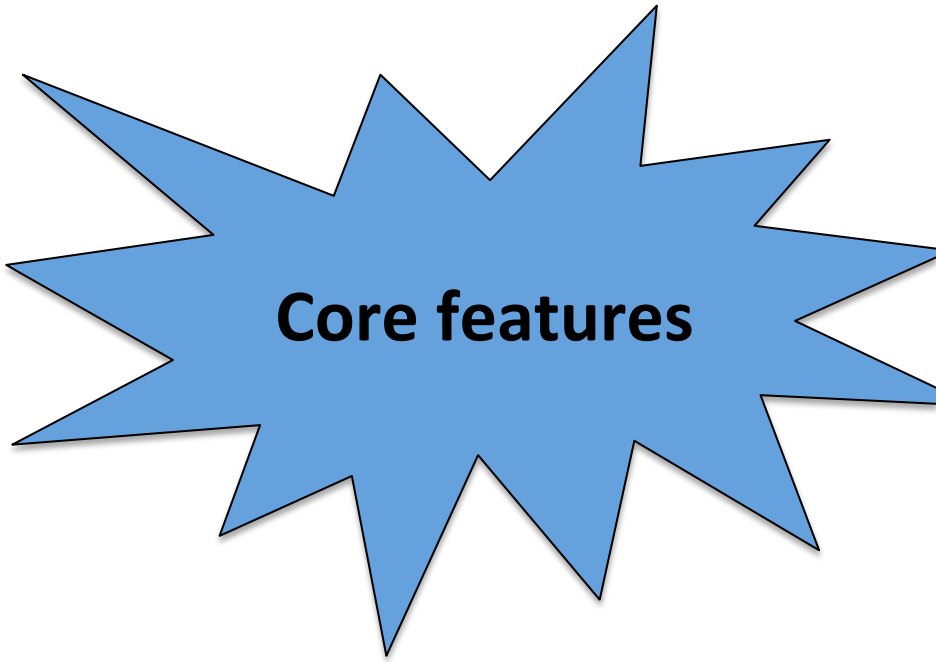
Optional

Xor-Group

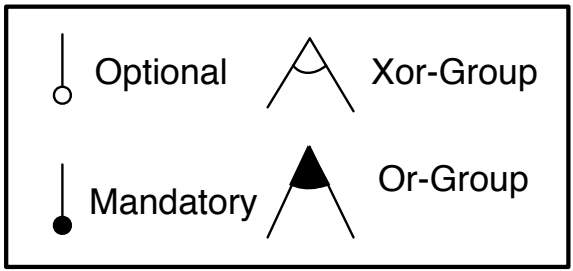
Mandatory

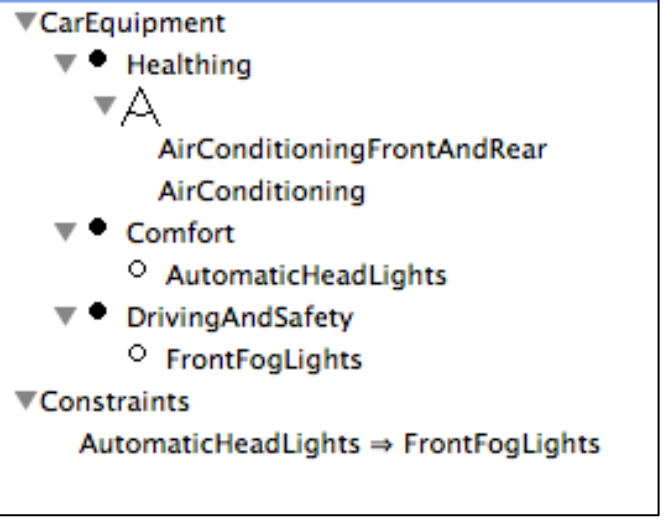
Or-Group

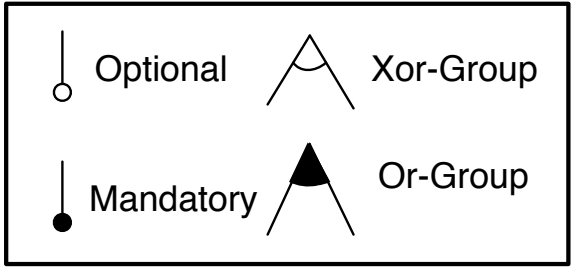


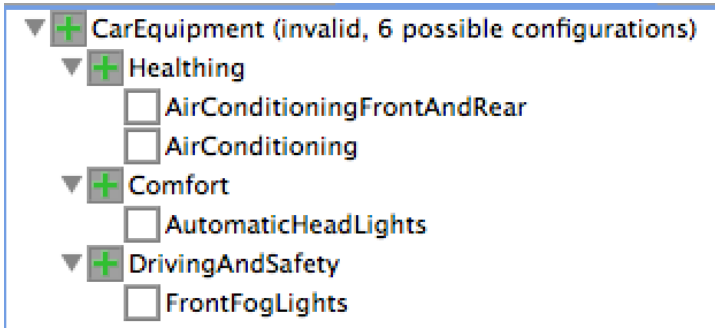


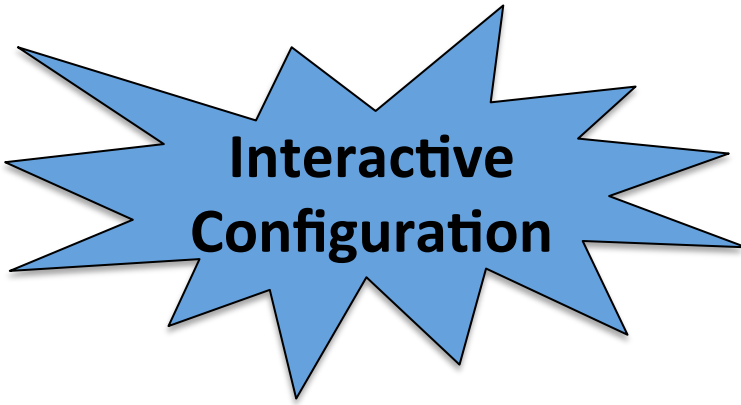
{CarEquipment, Comfort, DrivingAndSafety, Healthing}

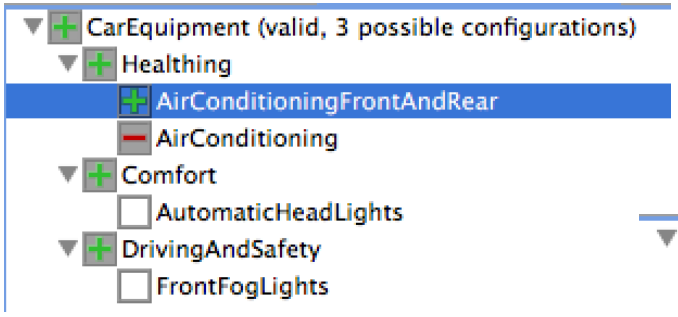


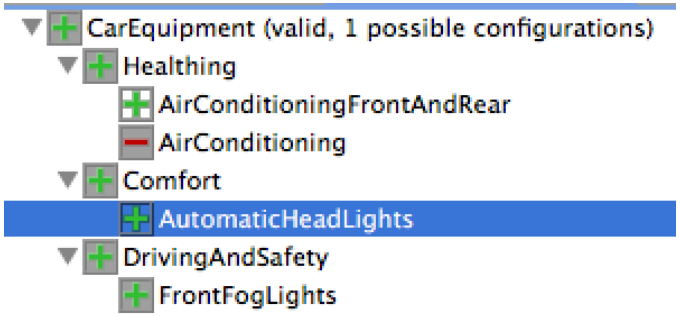

CarEquipment
Healthing
AirConditioningFrontAndRear
AirConditioning
Comfort
AutomaticHeadLights
DrivingAndSafety
FrontFogLights
Constraints
AutomaticHeadLights ⇒ FrontFogLights


Optional
Xor-Group
Mandatory
Or-Group


CarEquipment (invalid, 6 possible configurations)
Healthing
AirConditioningFrontAndRear
AirConditioning
Comfort
AutomaticHeadLights
DrivingAndSafety
FrontFogLights


Interactive
Configuration


CarEquipment (valid, 3 possible configurations)
Healthing
AirConditioningFrontAndRear
AirConditioning
Comfort
AutomaticHeadLights
DrivingAndSafety
FrontFogLights


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

Software Product Line Engineering

- Family of systems, configurable systems: paramount but hard to develop, test, and maintain
- Specific development process: domain engineering and application engineering
- Domain engineering: elicitation of variability requirements, commonalities, features, scoping
- Modeling requirements: domain model, feature model, product comparison matrix
  - Based on the analysis of textual artefacts (e.g., product descriptions), source code, knowledge, workshop, etc.
  - Models can automate the derivation and testing of variants

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