

# Software Product Line Engineering

Jean-Marc Jézéquel

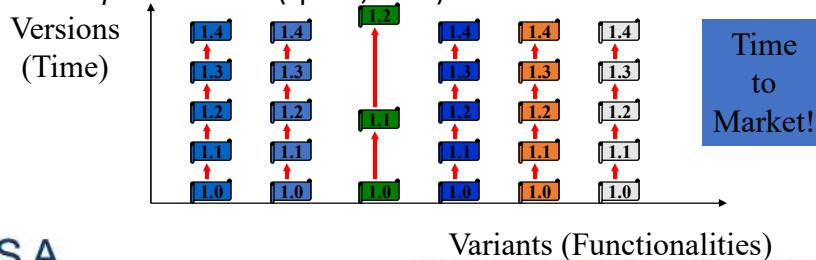
[jezequel@irisa.fr](mailto:jezequel@irisa.fr)

 @jmjezequel

## Modern Software Problems



- Importance of non-functional properties
  - distributed systems, parallel & asynchronous
  - quality of service : reliability, latency, performance...
- Flexibility of functional aspects: Product Lines
  - notion of *product lines* (space, time)







« 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 »

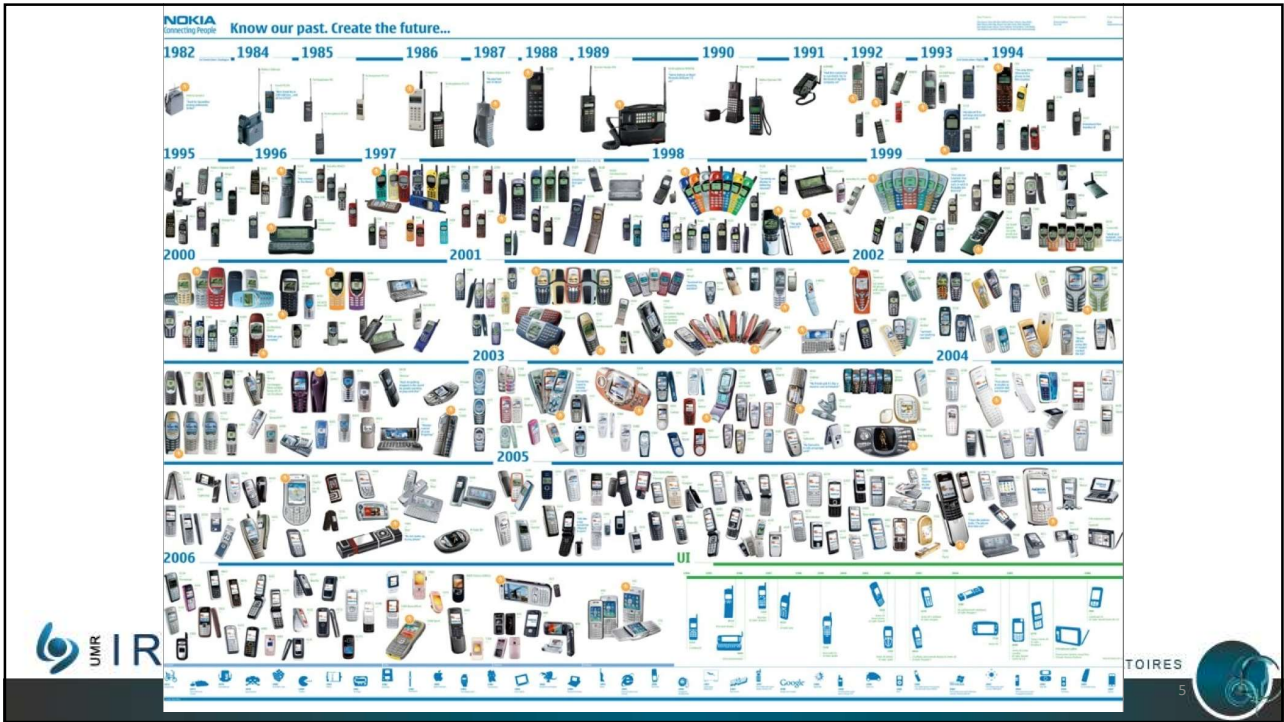
aka **Variability**

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



				
	Starter	Home Premium Upgrade \$119.99* <input type="button" value="Buy"/>	Professional Upgrade \$199.99* <input type="button" value="Buy"/>	Ultimate Upgrade \$219.99* <input type="button" value="Buy"/>
<b>Communication</b>				
Bluetooth support	✗	✓	✓	✓
Join a homegroup	✗	✓	✓	✓
Internet Explorer 8	✗	✓	✓	✓
View Available Networks	✗	✓	✓	✓
Windows Connect Now (WCN)	✗	✓	✓	✓
Create a homegroup		✓	✓	✓
Location and other sensors support		✓	✓	✓
Support for joining domains			✓	✓
<b>Entertainment</b>				
DirectX 11	✗	✓	✓	✓
Gadgets	✗	✓	✓	✓
Games Explorer	✗	✓	✓	✓
Play To	✗	✓	✓	✓
Windows Media Player 12	✗	✓	✓	✓
Create and play DVDs		✓	✓	✓
Intersense TV		✓	✓	✓





# Software-intensive systems

come in many variants

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES

The diagram illustrates the concept of software-intensive systems. On the left, a vertical stack of icons represents various software applications: a green Android robot, a penguin (Linux), a "text NOW" speech bubble, a map application, and an email icon. In the center, a large white arrow points from a grid of binary code (0s and 1s) to a central image of a Samsung smartphone. To the right of the smartphone, a vertical stack of six different mobile phones shows various user interface designs. At the bottom left is the UMR IR logo, and at the bottom right is the "ÉTOIRES" logo.

## Software Reuse

- drastically decrease cost of software development and maintenance
- increase quality of software
- reuse of existing software is one of the most promising approaches
- construct applications by composing reusable software pieces

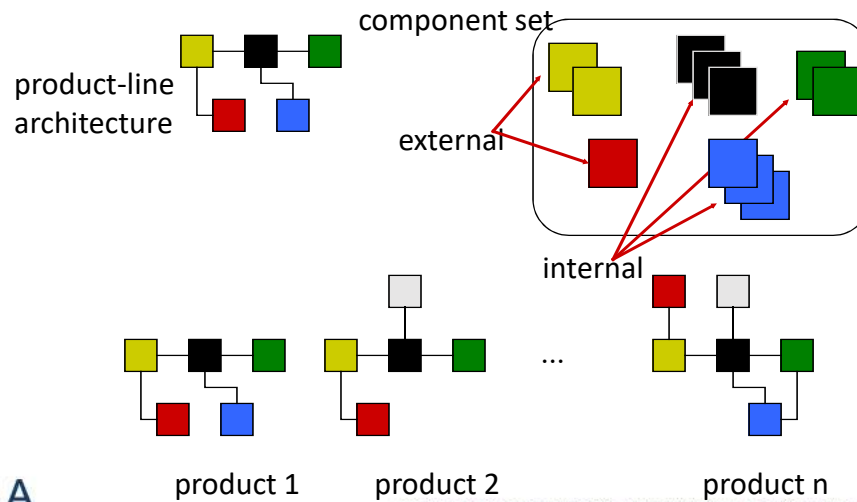


## Two approaches to reuse

- Opportunistic:
  - the software engineer reuses pieces of software that fit the current problem and adds them to the software.
- Planned:
  - the organization puts explicit effort in developing reusable artifacts that provide the 'right' abstractions, 'right' level of variability and that fit into an higher level structure.
- Opportunistic reuse does not work in practice
  - As in automotive and other industries, build on the notion of **Product Line**

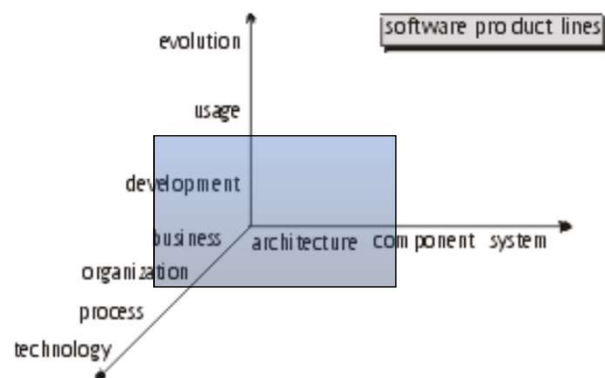


## Software Product Lines



## Many Issues Around SPLs

- Assets [Jacobsen]:
  - architecture
  - components
  - systems
- Views [SEI]:
  - business
  - organization
  - process
  - technology
- Lifecycle [Bosch]:
  - development
  - usage
  - evolution



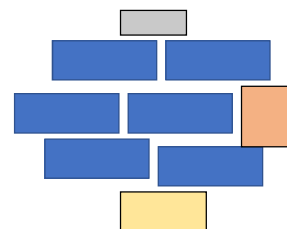
# Technical Issues in SPLs: Managing variability



## 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]





**RENAULT VANS**

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

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**

UMR IRISA INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES

## Benefits

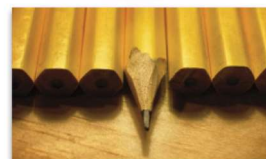
Improve product reliability



Improve usability



Improve consistency across products...





## Benefits

Reduce production costs



Reduce certification costs



Shorten time-to-market



## Hall of Fame

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



**PHILIPS**

**ERICSSON**



**NOKIA**  
Connecting People



**CelsiusTech**

**Lucent Technologies**  
Bell Labs Innovations



# Printer Firmware

invent

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



## Initiating a Product Line

	Evolutionary	Revolutionary
Existing product line	Develop vision for PLA Develop one comp. at a time by evolving existing components	Develop PLA, components and products based on requirement super set
New product line	PLA and components evolve based on requirements posed by new PL members	PLA and components developed to match requirements of all PL members

## Evolving Existing Products

- Advantages
  - reduced risk due to
    - small up front investment
    - early return on first shared components
  - relatively small effect on production schedule
- Disadvantages
  - larger total investment



## Replace Existing Products

- Advantages
  - shorter conversion time
  - smaller total investment
- Disadvantages
  - higher risk
  - negative effect on production schedule

Warning: role of hardware and mechanical parts

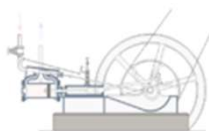


## Applicability of SPL concepts

- context: consultancy company or IT department performing projects with partially overlapping requirements
- define common architecture and common components/subsystems
- develop (slightly) more general components in normal projects
- increase the variability and generality of the component in subsequent projects
- **balance investment and risk!**



## A Bit of History: Industrial Revolution



1698  
Thomas Savery



1901  
Henry Ford



1980s



# Nowaday: Product Lines Everywhere



## LE PLIAGE PERSONNALISÉ

MAISON

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



# Product Lines of Cars



Exterior | Interior | Side | Front | Rear

1. Trims/Series 2. Engine/Transmission 3. Colour & Style 4. Options 5. Summary

### Choose Your Options

Audio/Comms/Nav Heating/Ventilation Mechanical Safety/Security A-Z

**Audio/Comms/Nav**

- CD 30** - MP3 CD player with MP3 format, stereo radio, steering wheel mounted audio controls **Standard**

**Heating/Ventilation**

- Air conditioning** **€ 923.00**

**Mechanical**

- Electronic Stability Programme (ESP)** **€ 411.00**

**Safety/Security**

- Emergency tyre inflation kit in lieu of space-saver spare wheel and tyre** **Standard**

**Pricing Details**

Club	€ 14,350.00
1.2i 16v, 5 Speed	
Blaze Red	€ 0.00
Melt / Elba Charcoal	€ 0.00
15-inch steel wheels with 185/60 R 15 tyres and flush wheel covers	€ 0.00
<b>Total</b>	<b>€ 15,684.00</b>

**Options (2)**

You selected:

- Air conditioning € 923.00
- Electronic Stability Programme (ESP) € 411.00

**Total** € 15,684.00

**Next Step: Summary**

**Legend**

- Selected Option
- Selectable Option
- Option contained in an option pack
- Option contained in an option pack or standard equipment which has been replaced by another option
- Option that is only selectable together with another option. Please click for details



Willkommen bei selve - the shoe individualizer

http://www.selve.net/index\_js.html

KOLLEKTION FUßSTY P HYSELVE INFO HOME

MODELLE LOOKBOOK SELVE ID PASSWORD ANMELDEN

selve Kollektion -> Style: casuals -> Modell: Opal

modell-details >> hier klicken

>>SELVE SCHUHEGAL Inhalt:0

>>SHOPPING BAG Inhalt:0

A. Erstes Oberleder: **Veloursleder Sand**

B. Veloursleder Bordeaux: **Veloursleder Cognac**

C. Veloursleder Sand: **Veloursleder Sand**

D. Absatz: **Hufeisen Braun**

E. Sohle: **Gummisohle**

ÄNDERN ZURÜCKLEGEN

**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  
Ort: Konstantzelle DE-937  
Impressum

UMR IRISA

Der Dell Online-Shop: Stellen Sie Ihr eigenes System zusammen - Mozilla Firefox

http://configure2.euro.dell.com/dellstore/config.aspx?c=de&cs=dedhs1&kc=3058l=de&oc=W06390xp&ss=dhs&dbc=pr

Bestellen Sie online oder wählen Sie 0800 533 55 40 (gebührenfrei)

**DELL** Produkte Service Support Einkaufsunterstützung

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 Warenkorb

**Als Symbol anzeigen**  
EUC-BUNZ-SURKAB-Speicher mit 40 GB und 607 MHz (2 x 20 GB DIMM) [plus 0,19,99 € oder 2 € / Monat]

**Grafikkarte**  
128 MB nVidia NVS285 DVIA/GA-Grafikkarte

Auswahlhilfe

- 256 MB ATI Fire GL V7200-Grafikkarte [plus 416,50 € oder 13 € / Monat]
- 128 MB nVidia Quadro FX550-Grafikkarte [plus 69,02 € oder 2 € / Monat]
- 256 MB nVidia Quadro FX3450-Grafikkarte [plus 547,40 € oder 17 € / Monat]
- 128 MB nVidia NVS285 DVIA/GA-Grafikkarte [im Preis enthalten]
- Grafikkarte PCIe x16 (DVIA/GA) Matrox D1D LP PCIe, 128 MB, DVI- oder VGA-Grafikkarte für 4 Monitore [plus 630,70 € oder 20 € / Monat]
- 128 MB ATI Fire GL V3400-Grafikkarte [plus 44,03 € oder 1 € / Monat]

**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)**  
inkl. MwSt., zzgl. 19,04 € Versand  
\*\*Ernäßigter Sonderpreis\*\*  
**913,92 €**  
Es gelten keine zusätzlichen Preisnachlässe. Das Angebot gilt für maximal 5 Systeme

Finanzierung ab 30 €/mtl.<sup>2</sup>  
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".

UMR IRI

Transferring data from i.dell.com...

# Food? Product lines!



**VEGETARIAN**

WHICH WICH WOULD YOU LIKE?

TRIPLE CHEESE MELT  
 ELVIS WICH (p. Ham & Banana)  
 TOMATO & AVOCADO  
 BLACK BEAN PATTY  
 HUMMUS & BELL PEPPERS

CHOOSE YOUR BREAD

WHITE  WHEAT

CHOOSE YOUR CHEESE (By bread)

AMERICAN  SWISS  PROVOLONE  
 CHEDDAR  PEPPER JACK  MOZZARELLA

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

MUSTARDS  
 Yellow  Dijon  Honey  Deli

MAYOS  
 Regular  Lite  Horseradish  Spicy

SPREADS & SAUCES  
 BBQ  Buffalo  Marinara  
 1000 Island  Ranch

ONIONS  
 Red  Grilled  Crispy Strings

VEGGIES  
 Lettuce  Tomato  Pickles  Jalapenos  
 Olive Salad  Mushrooms  Spicy Ketchup  
 Kalelaw  Bell Peppers

OILS & SPICES  
 Oil  Vinegar  
 Salt  Pepper  Oregano  Parmesan

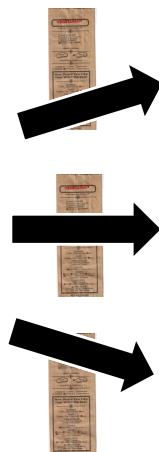
EXTRAS (75¢ Each)  
 Bacon  Avocado  Pickle (Whole)  
 Mars Meat  More Cheese

WRITE YOUR NAME HERE



INS

QUE ET SYSTEMES ALÉATOIRES



INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYS



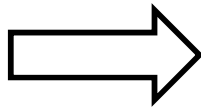


What about  
software?

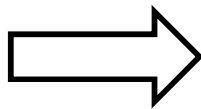


Product lines of  
software intensive systems

**Software** intensive systems  
are declined in many **variants**



**Software** intensive systems  
are declined in many **variants**



# Software Product Lines



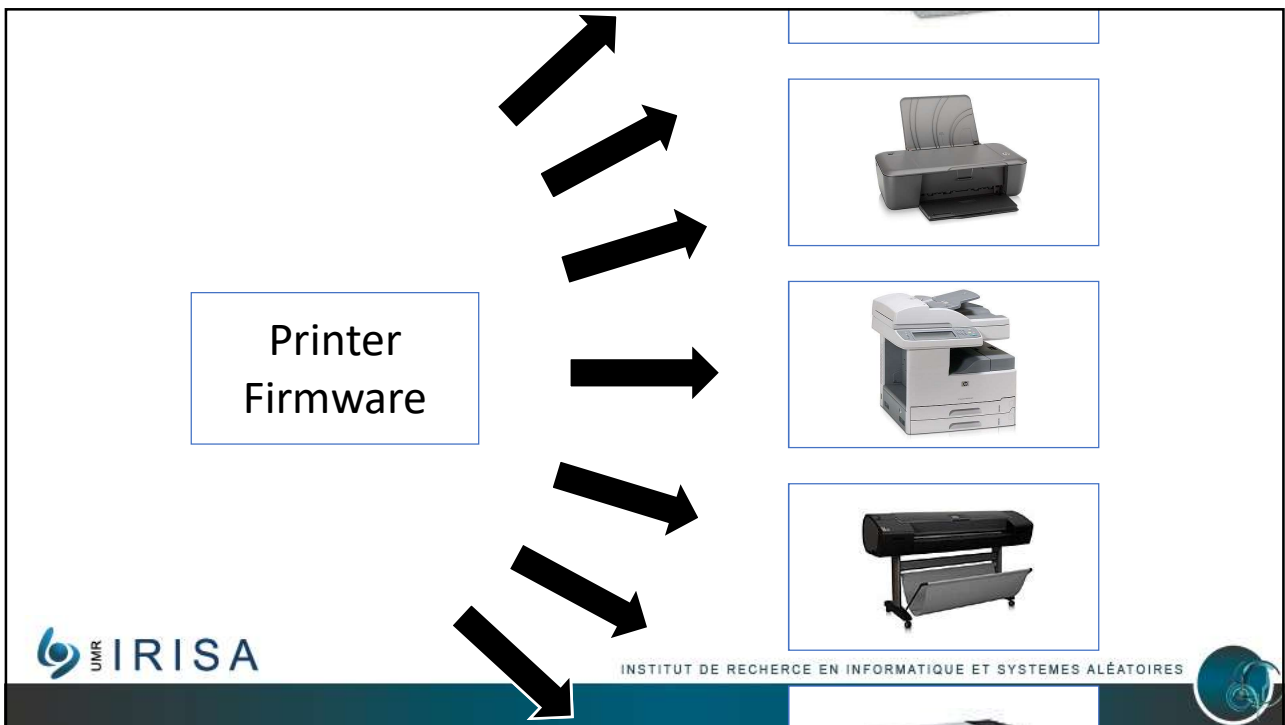
IN

CE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES

Car



INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



The diagram illustrates the application of the Linux Kernel to various hardware devices. On the left, a terminal window displays the configuration options for the Linux Kernel (version 2.6.33.3). Below the terminal is a box labeled "Linux Kernel" with the Tux penguin logo. Five black arrows point from the terminal area to a vertical stack of four images on the right: a server rack, a blue wireless router, and a smartphone. The bottom of the slide features the IRISA logo and the text "INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES".

## Features in Microsoft Office

The screenshot shows the "Microsoft Office 2003 Setup" window for "Microsoft Office Professional Edition 2003" in "Advanced Customization" mode. The "Choose installation options for applications and tools" section is expanded to show "Proofing Tools". Under "Proofing Tools", the following options are listed with checkboxes:

- New and Open Office Document Shortcuts
- Office 2003 Web Components
- English
- French
- Spanish
- Microsoft Office Download Control
- Themes
- Visual Basic for Applications

The "Description" for Proofing Tools is "Tools for proofreading Office documents." At the bottom right, it indicates "Space Required on C: 342 MB" and "Space Available on C: 3553 MB". Navigation buttons for "Help", "< Back", "Next >", and "Cancel" are visible at the bottom.

## Variability

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

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



**Variability = Complexity**



33 optional, independent features



a unique variant for every  
person on this planet

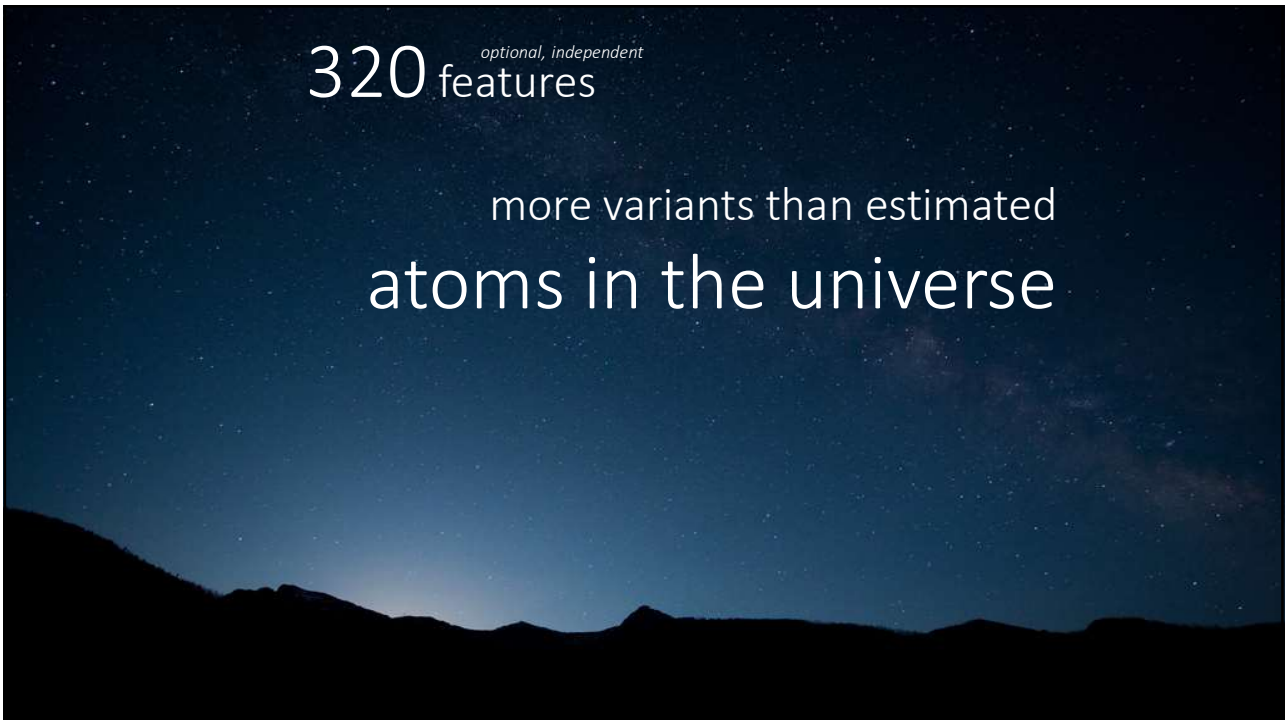




INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



320 optional, independent features


more variants than estimated  
atoms in the universe



**2000** features

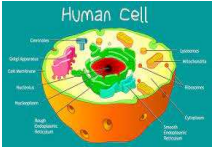
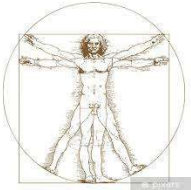


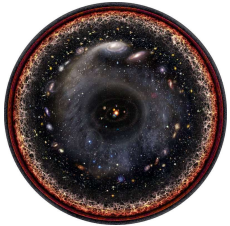
**15000** features



Yielding a *huge* number of possible configurations for a Linux Kernel  
 $2^{15000} \approx 10^{5000}$

UMR IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES

$10^{14}$        $10^{28}$        $10^{38}$        $10^{50}$        $10^{80}$

logarithmic scale

- 1 metre  $10^0$  m
- 1 millimetre  $10^{-3}$  m
- 1 micrometre  $10^{-6}$  m
- 1 nanometre  $10^{-9}$  m
- 1 picometre  $10^{-12}$  m
- 1 femtometre  $10^{-15}$  m
- 1 attometre  $10^{-18}$  m
- 1 zeptometre  $10^{-21}$  m
- 1 yoctometre  $10^{-24}$  m
- $10^{-27}$  m
- $10^{-30}$  m
- $10^{-33}$  m
- $10^{-36}$  m

person, blood cell, atom, atomic nucleus, Planck scale

Plank length:  $10^{-35}$  m  
 Volume of the observable Universe:  $10^{80}$  m<sup>3</sup>  
 => #places in Universe:  $10^{115}$

#configurations of Linux Kernel:  $10^{5000}$   
 #universes to store configurations of Linux Kernel:  $10^{43}$

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES



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

Reuse

Customization

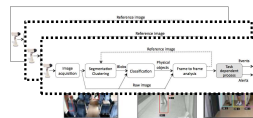
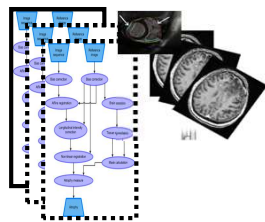
Automation

Commonality

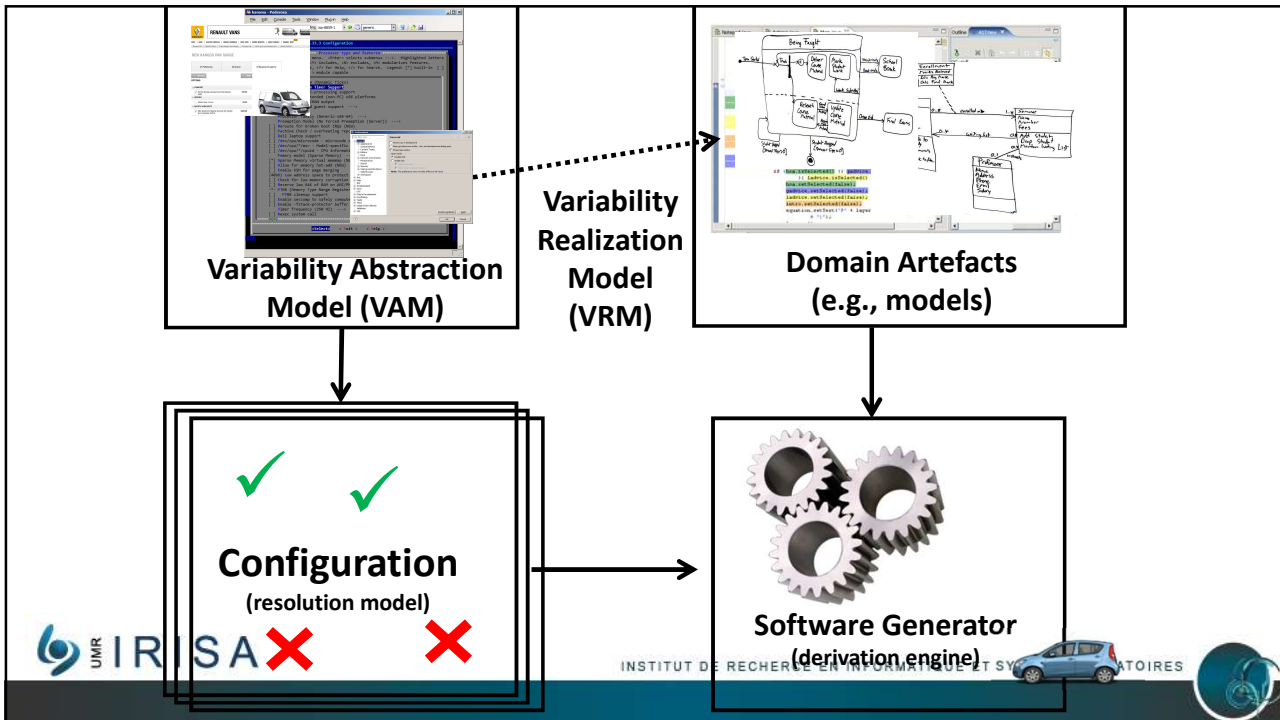
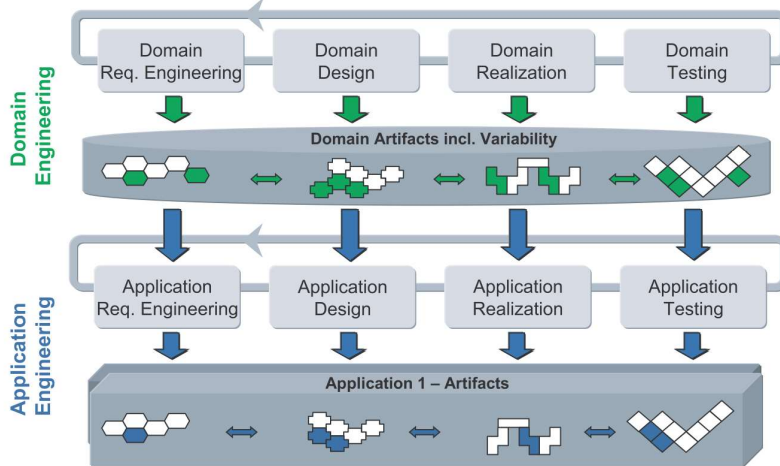
Variability

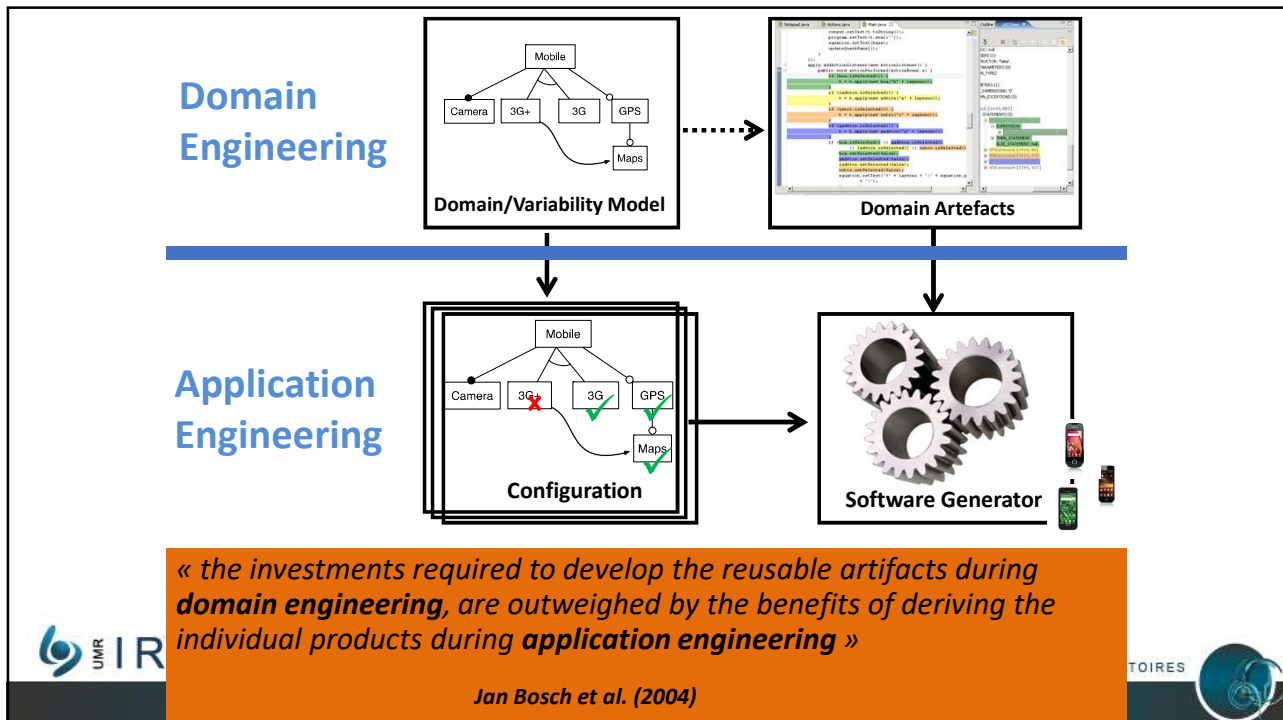
*“a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way” [Clements et al., 2001]*

## Software Product Lines



# Software Product-Line Engineering





## Variability Management, the Good Old Way

- Conditional compilation, e.g. in C with CPP

```
#IFDEF feature
xxx
#ELSE
YYY
#ENDIF
```

- Hard to manage, hard to change mind of what is static/dynamic
  - J.-M. Jézéquel. *Reifying Configuration Management for Object-Oriented Software*. ICSE 1998
- Recall: Linux Kernel has 15000 such « features »

## Inheritance (OOP)

Base Class encapsulate commonalities

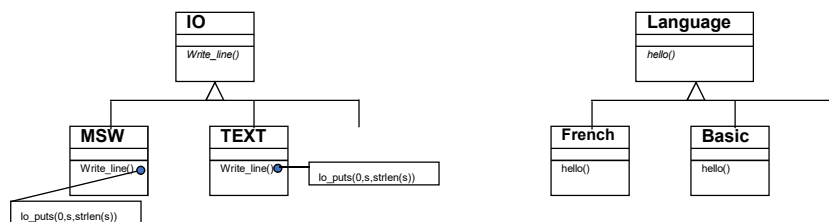
Derive classes specialize peculiarities



## Basic Idea



- Abstract the Intent
  - `io.write_line(language.hello)`
- Rely on Dynamic Binding for the Details
  - Don't care now for static/dynamic distinction

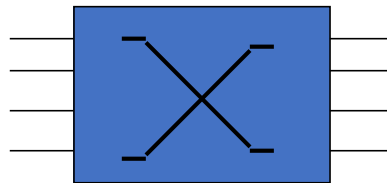


- Uncouple the variations from the selection process
  - Automatically derive a product using OCL 2 meta-model transformation



## The Mercure PL as example

- A family of SMDS<sub>[Jézéquel 96]</sub> (Switched Multi-Megabits Data Service) servers.
- Delivering, forwarding, and relaying messages from and to a set of network interfaces.



## Variability in UML class diagrams

### Abstraction

Inheritance, Abstract Factory

- **Parameterization**

UML class templates

- **Optionality**

UML extensions mechanisms (Stereotype « Optional »)

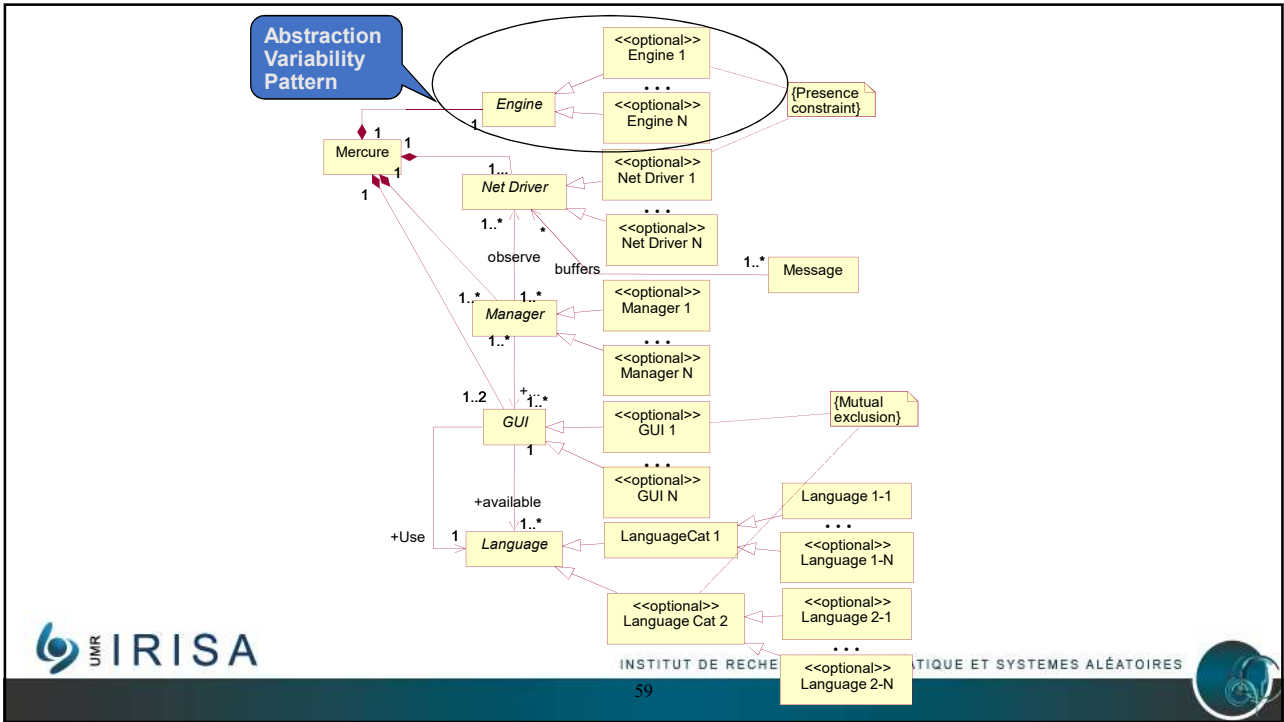
- **Alternatives**

Optional + Constraint

Tewfik Ziadi, Jean-Marc Jézéquel:

*Software Product Line Engineering with the UML:  
Deriving Products.* Software Product Lines 2006





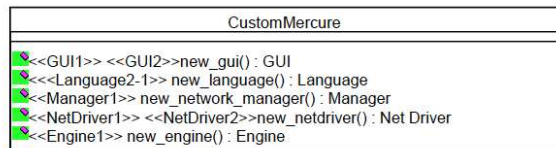
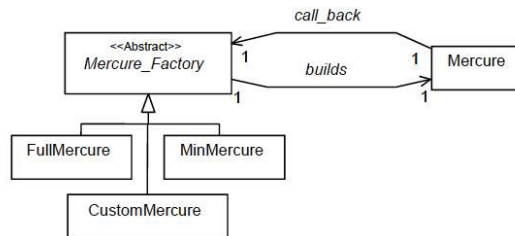
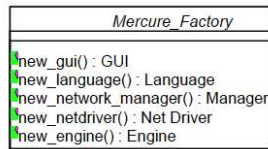
## Product Line Derivation

Products \ VPs	GUI	Language	Manager	NetDriver	Engine
CustomMercure	GUI1, GUI2	Language 2-1	Manager1	NetDriver1	Engine1
MiniMercure	GUI1	Language 1	Manager1	NetDriver1	Engine1
FullMercure	all	all	all	all	all

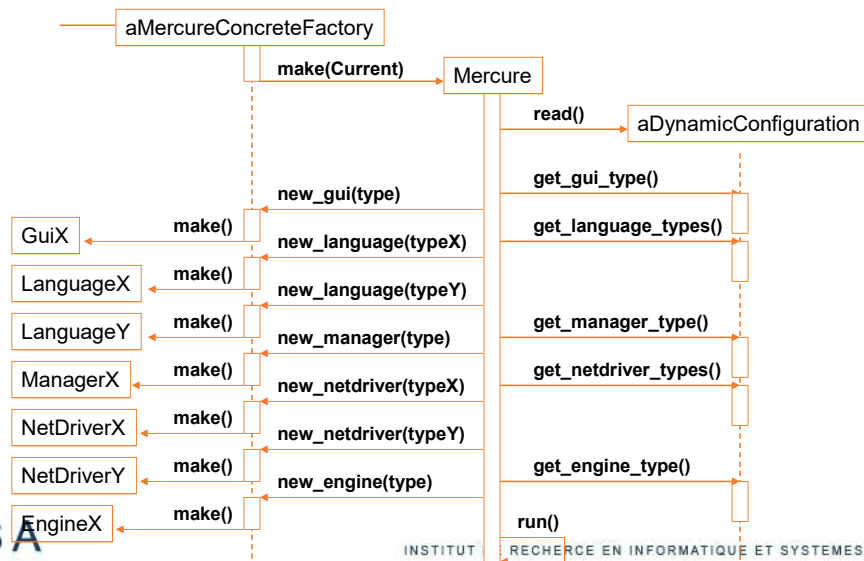
GUI 1 *mutual-exclusion* Lang Cat 2

## The decision model

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



## Dynamic Configuration



## The PL derivation

- Manipulate the UML meta-model to automatically derive a product model using a Model Transformation Language (MTL)

### DeriveProductLine

**Input:** **PL\_model:** Model

**aConcreteFactory:** Class

pre : -- check Generic Constraints on PL\_model

**Output:** **Product\_model:** Model

post :-- check Specific Constraints on the PL\_model



## Model Transformation

- By limiting the range of variants available from a given Concrete Factory:
  - The transformer may know the set of *living* classes
    - special case of Partial Evaluation
  - Generate a **specialized** model for the product
    - When only one *living* class for an abstract varying part:
      - Dynamic binding replaced with direct call (and even inlining)
    - When only a few *living* classes
      - Dynamic binding replaced by *if then ... else*
      - Implemented in e.g., GNU SmallEiffel
- All static configuration issues kept encapsulated and do not pollute the model





## The PL derivation

- The Variants selection:
  - Using operation factory stereotypes
- The Model specialization:
  - Removes all optional classes which have not been selected
- The Model optimization:
  - Deletes unused factories, Optimize inheritance



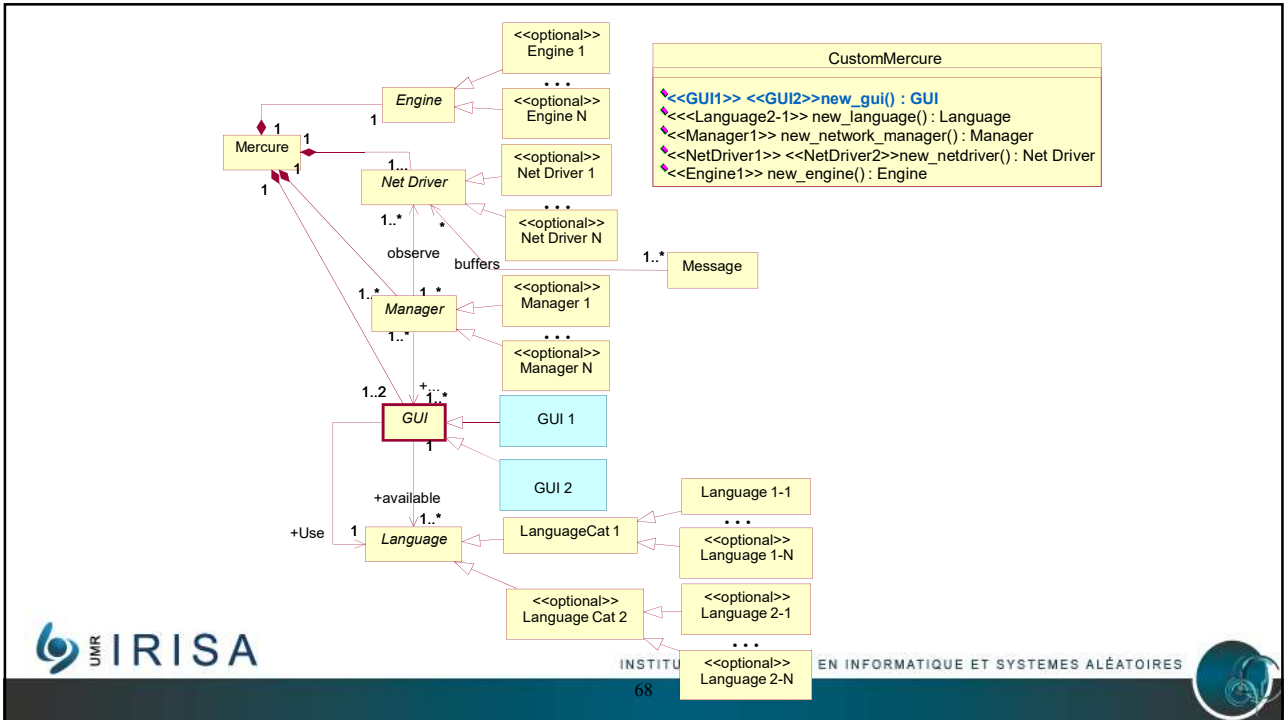
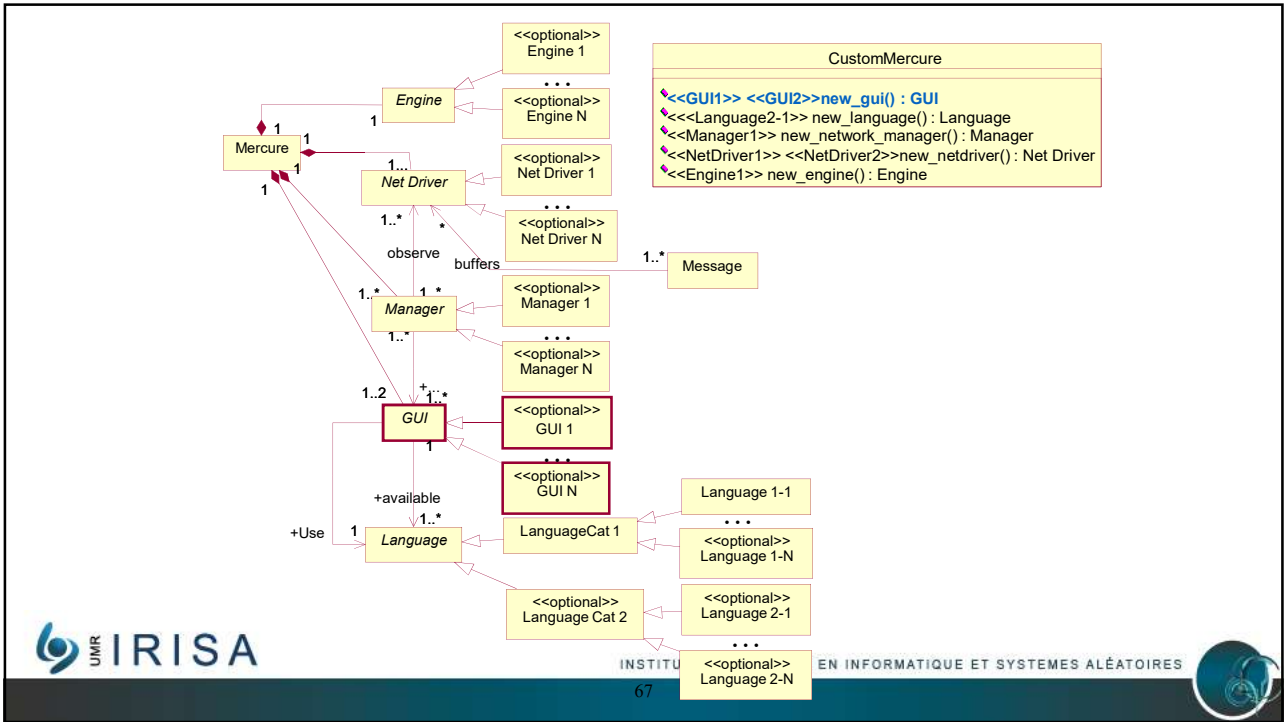
```

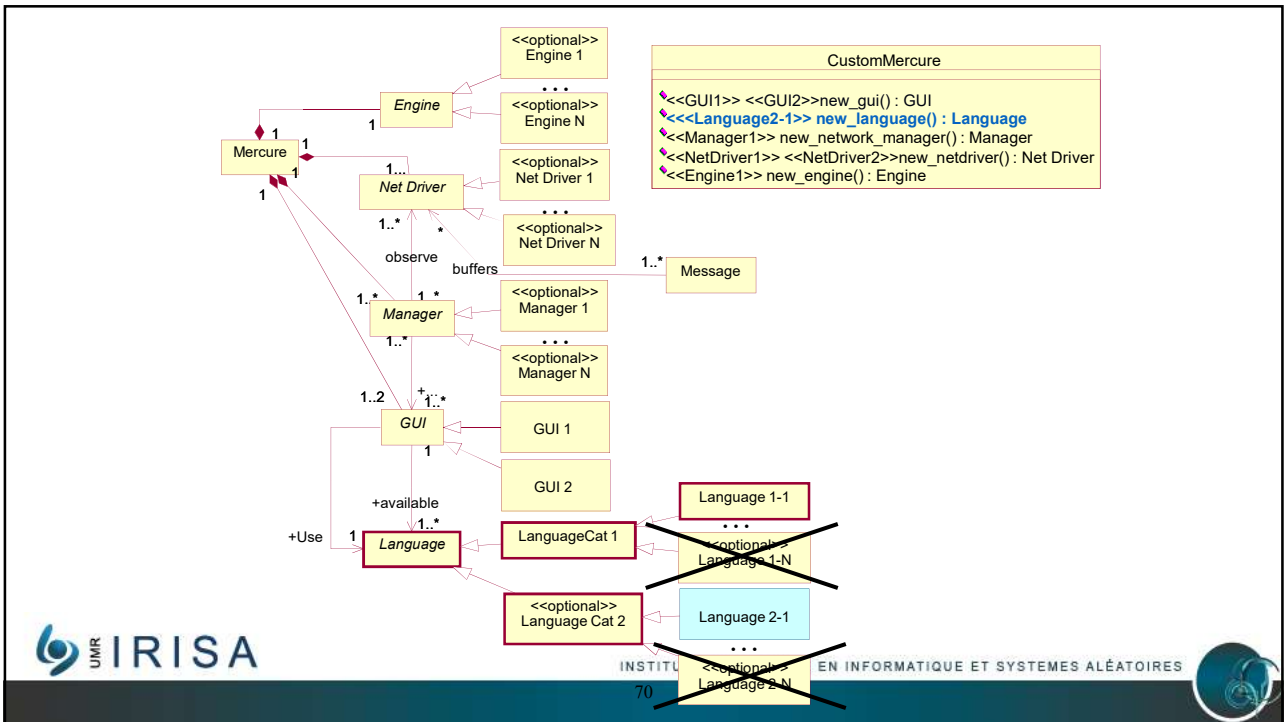
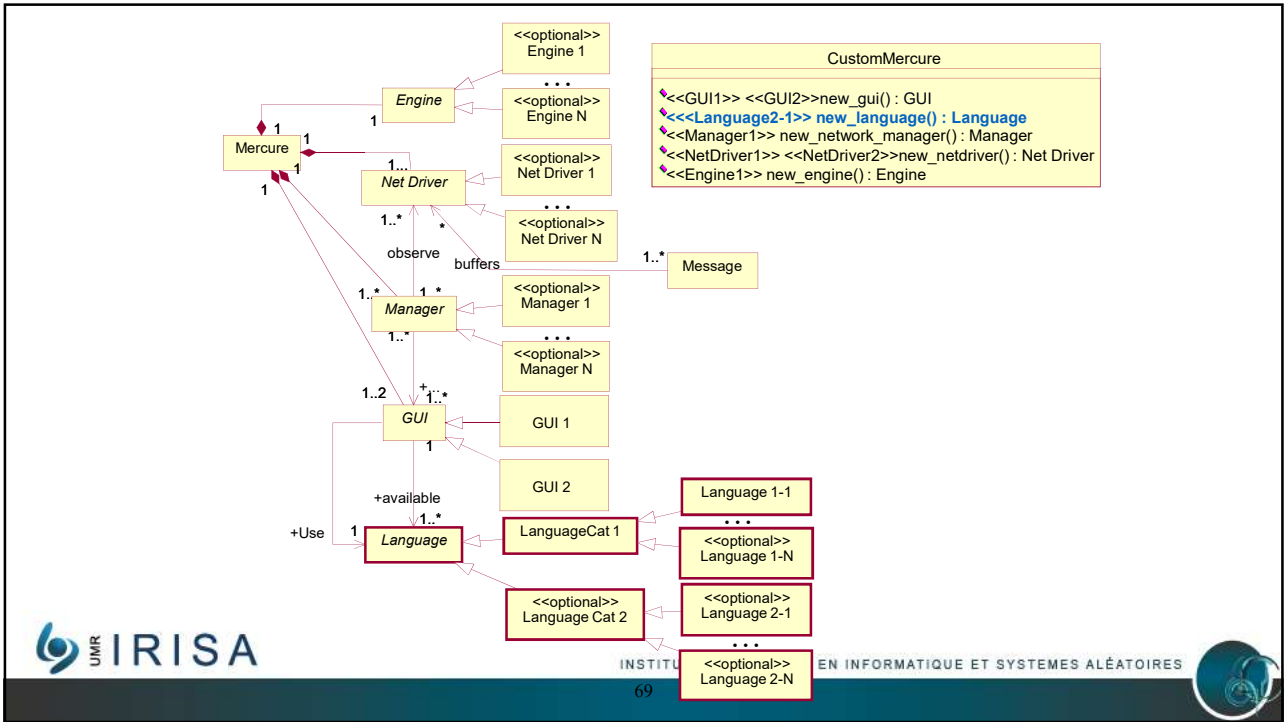
DerivePL (PL_model: Model, aConcreteFactory: Class) : Model {

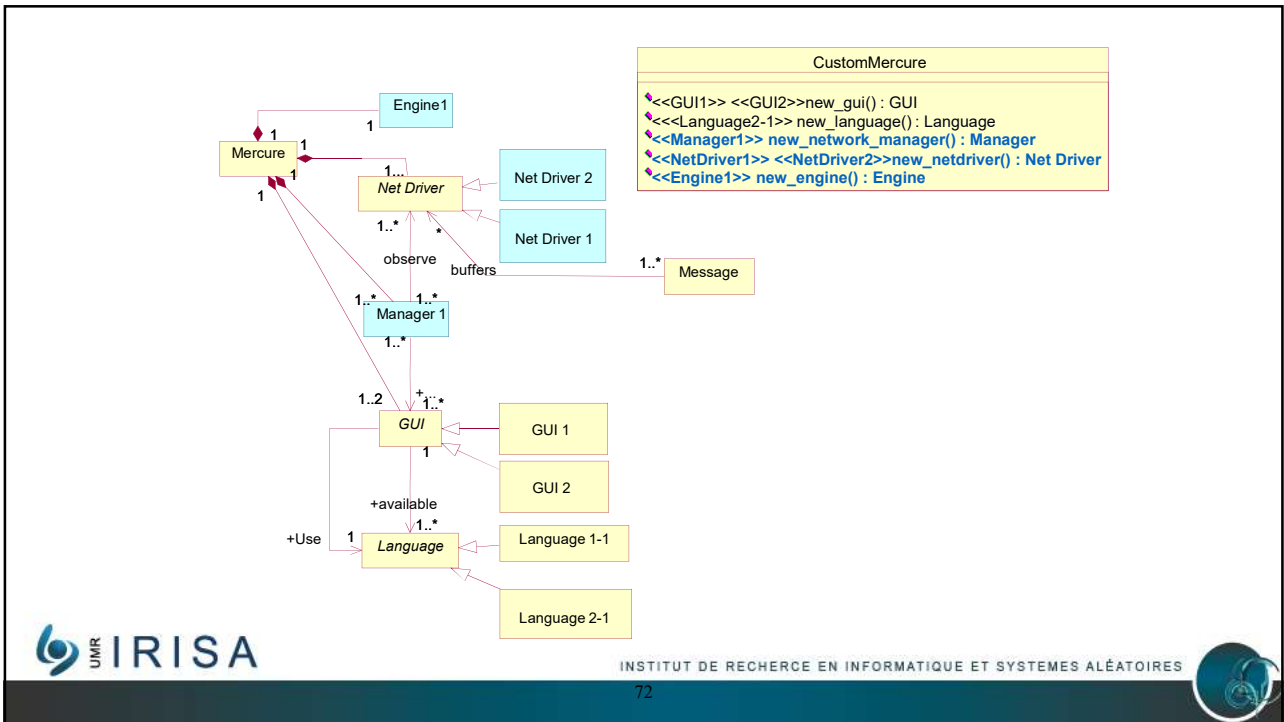
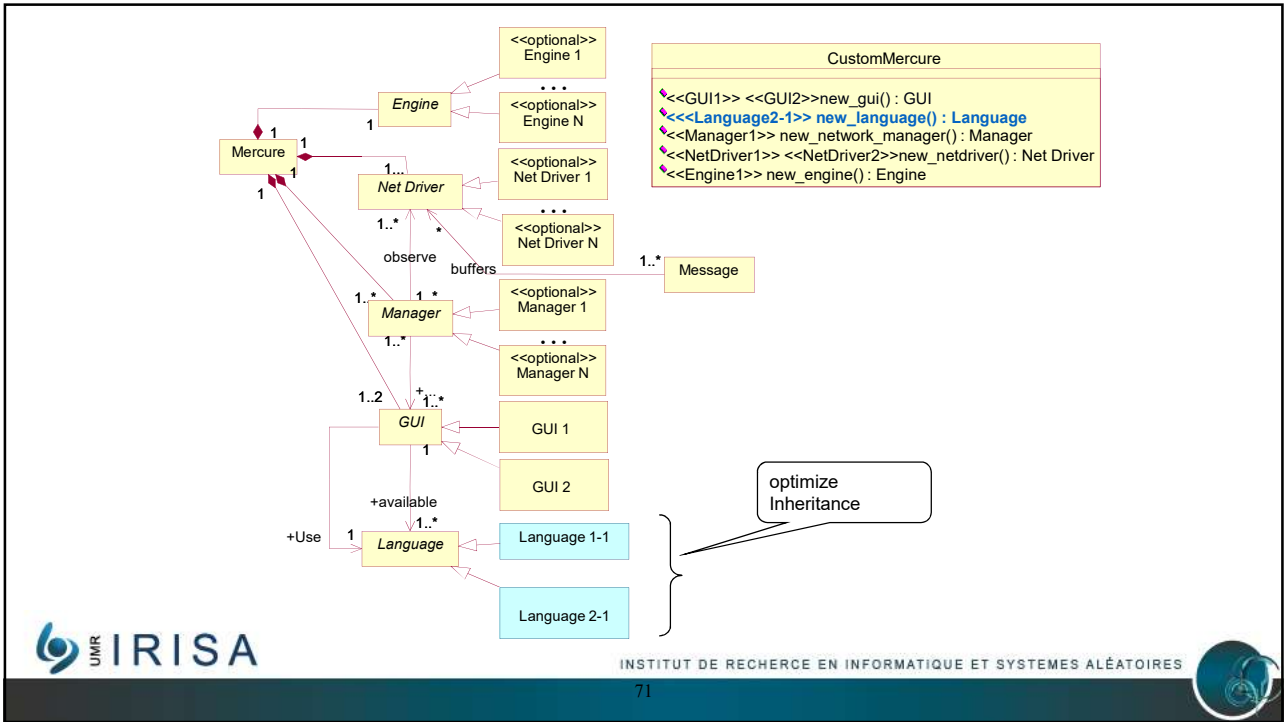
    selectedVariants: Set;
    Result := clone(PL_model);
    selectedVariants := getSelectedVariants(aConcreteFactory);
    // Model specialization
    for each optional class C in PL_model do
        if (the class name of C not in selectedVariants)
            and names of all subclasses of C not in selectedVariants)
        then
            delete the class C from Result;
        endif
    done
    // Model optimization
    replace abstract classes with only one subclass S by S
    delete all other factories
}

```

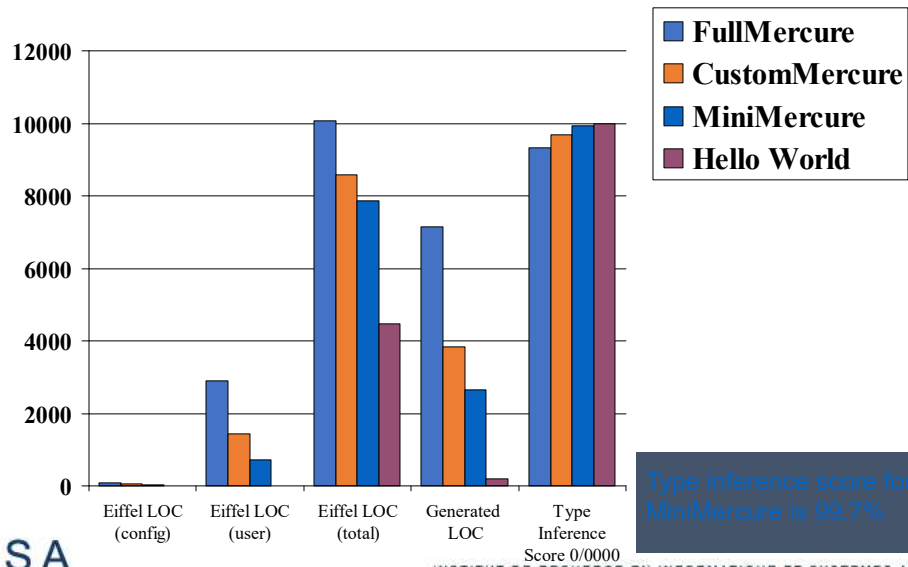




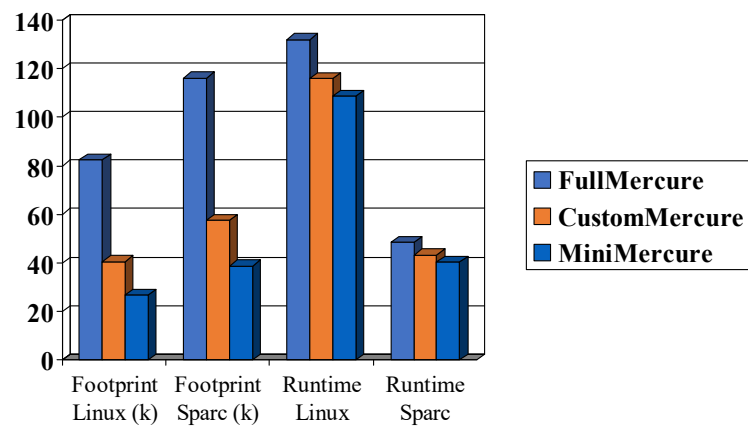




## Code size



## Runtime Performances



API

Framework



Plugin-based systems



(Active) Annotations  
can have parameters



Metamodeling and Domain-Specific  
Languages



### httpd.conf -- win32 Apache

Building a Web Server, for Windows

```

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

ServerName localhost:80
ServerAdmin admin@localhost

ServerSignature On
ServerTokens Full

DefaultType text/plain
AddDefaultCharset ISO-8859-1

UseCanonicalName Off

HostnameLookups Off

ErrorLog logs/error.log
LogLevel error

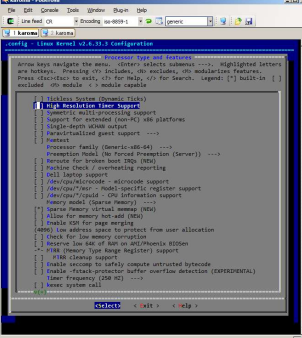
PidFile logs/httpd.pid

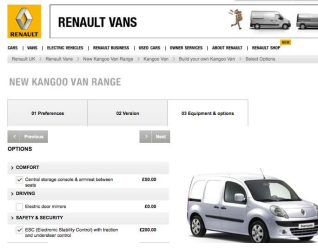
Timeout 300

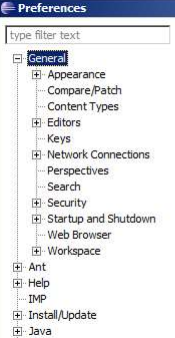
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

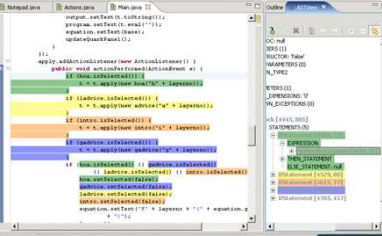
<IfModule mpm_winnt.>
  ThreadsPerChild 250
  MaxRequestsPerChild 0

```









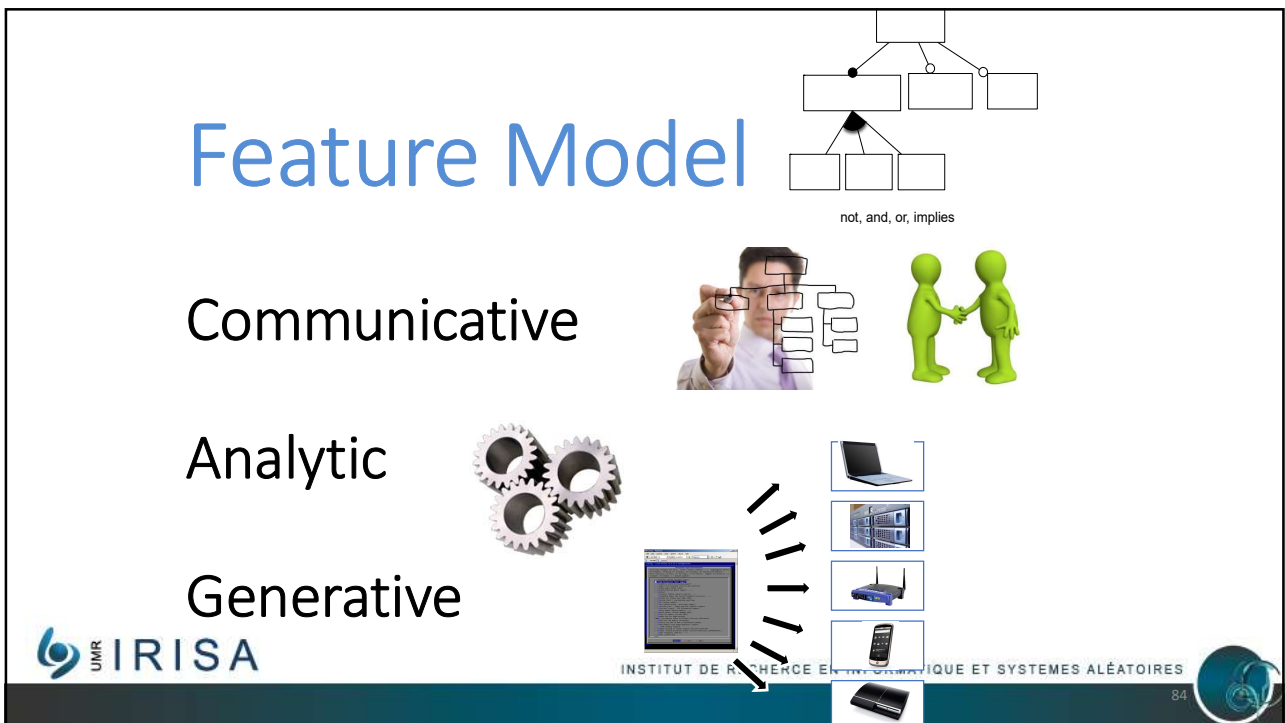
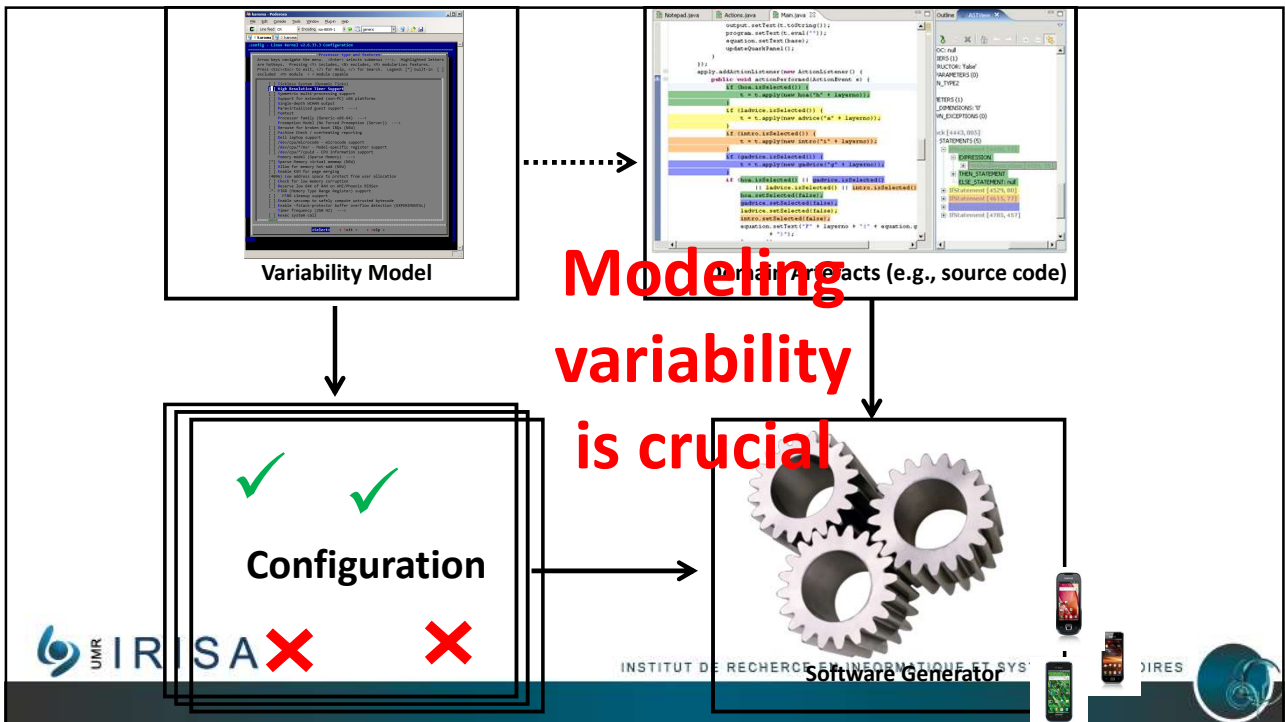




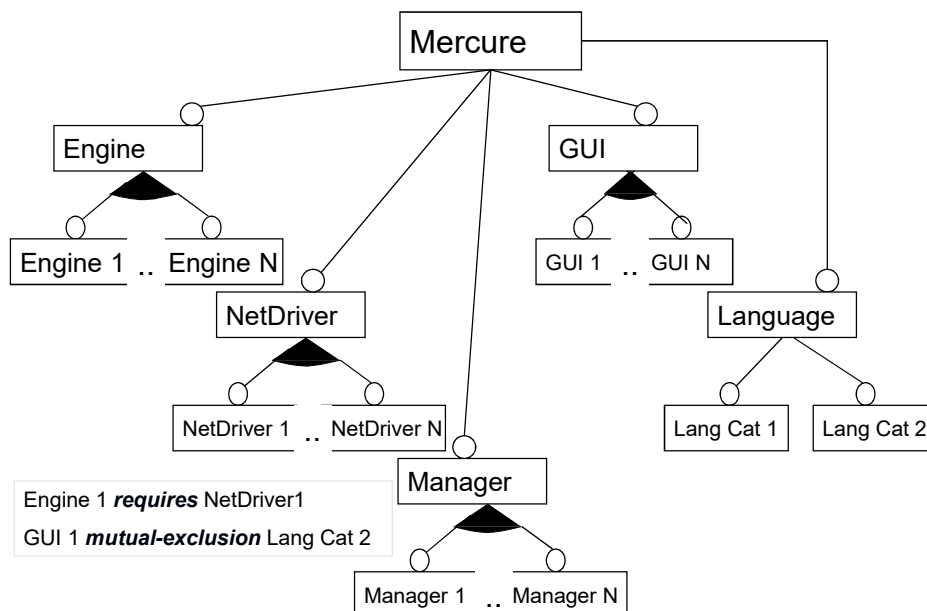
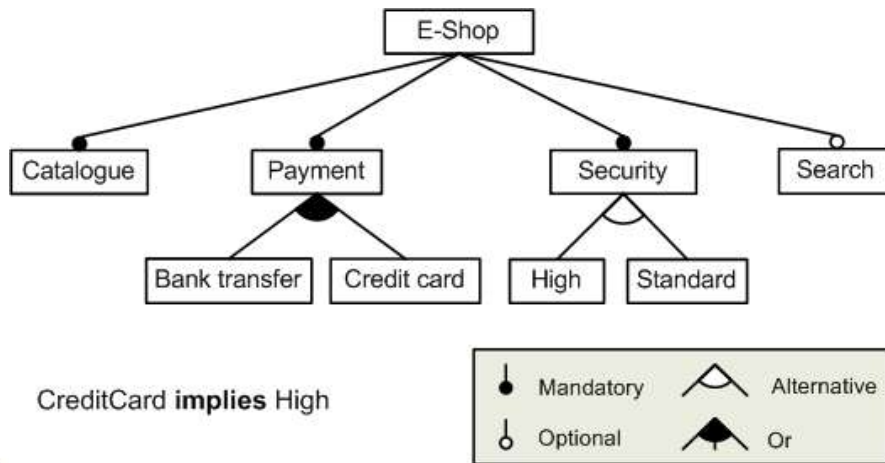



# Feature Models





## Feature Modeling





Vorsprung durch Technik Audi

### R8 Spyder

5.2 FSI quattro R tronic

**Prix total**  
171.216,00 EUR

Prix de base 170.490,00 EUR  
Équipements optionnels 726,00 EUR

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

Plein écran / Dimensions Fermer la capote Habitacle Tableau de bord

**Packs**  
Aucun pack n'est proposé pour ce modèle.

**Couleurs**  
Blanc Ibis  
Noir  
Prix: 0,00 EUR

Couleurs métallisées à partir de 0,00 EUR


Couleurs à effet perlé à partir de 0,00 EUR

Couleurs personnalisées Audi exclusive


**Couleur capote**  
Noir


**Jantes**  
4 Jantes alu 5 BRANCHES ROTOR finition titane 8,5 x 19 à l'avant, 11 x 19 à l'arrière. Pneus 235/35 R19 à l'avant et 305 /30 R19 à l'arrière  
Prix: 726,00 EUR

19" à partir de 0,00 EUR



1 Modèle 2 Moteur 3 Extérieur 4 Intérieur 5 Option 6 Votre Audi





Vorsprung durch Technik Audi

### R8 Spyder

5.2 FSI quattro R tronic

**Prix total**  
185.899,35 EUR

Prix de base 170.490,00 EUR  
Équipements optionnels 15.409,35 EUR

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

Plein écran / Dimensions Vue extérieure Tableau de bord


- Packs d'équipements
- Extérieur
- Jantes & pneumatiques
- Intérieur
- Volants
- Sièges
- Sécurité & technique**
- Infotainment

- Châssis
- Freins
- Systèmes d'assistance**
- Autres


**excludes**

- Régulateur de vitesse 320,65 EUR
- Système d'aide au stationnement APS avant / arrière 931,70 EUR
- Système d'aide au stationnement APS avant / arrière avec affichage dans l'écran MMI 1.373,35 EUR
- Système d'aide au stationnement Advanced : APS avant et arrière et caméra arrière 1.790,80 EUR
- Audi hill assist : assistance au démarrage en côte Série

Réinitialiser la sélection



1 Modèle 2 Moteur 3 Extérieur 4 Intérieur 5 Option 6 Votre Audi



Vorsprung durch Technik Audi

**A5 Sportback**  
3.0 TDI quattro S tronic

Prix total  
54.460,15 EUR

Prix de base 50.570,00 EUR  
Équipements optionnels 3.890,15 EUR

GPS Plus avec disque dur 2.934,25 EUR  
 Ordinateur de bord en couleur avec programme efficiency 181,50 EUR  
 Pack Intenso Plus 3.100,00 EUR

**Vérification de votre sélection**

Cet équipement nécessite un équipement complémentaire:

**Voici les équipements complémentaires possibles:**

Remarque: uniquement sur les modèles avec système Start-Stop et uniquement disponible en combinaison avec l'autoradio Concert, l'autoradio Symphony ou un système de navigation

1 Modèle 2 Moteur 3 Extérieur 4 Intérieur 5 Option 6 Votre Audi

UMR IRISA

## Feature Models

IDEA

- CarEquipment
  - Heating
    - AirConditioningFrontAndRear
    - AirConditioning
  - Comfort
    - AutomaticHeadLights
    - DrivingAndSafety
    - FrontFogLights
  - Constraints
    - AutomaticHeadLights => FrontFogLights

UMR IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTEMES ALÉATOIRES

90

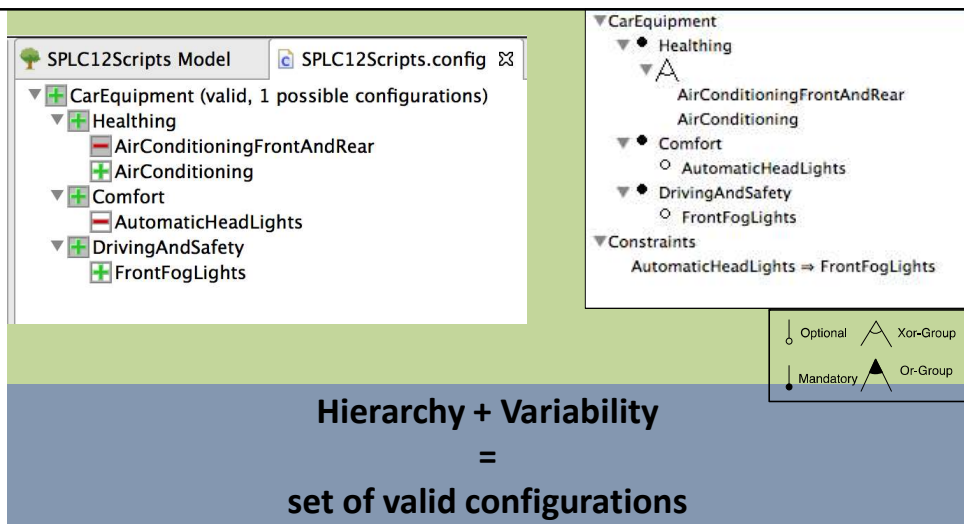
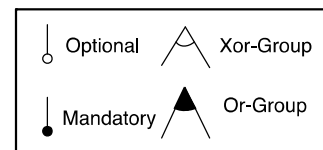
# Feature Models (Background)



**Hierarchy:** rooted tree

**Variability:**

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



**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

CarEquipment

- Healthing
  - Xor-Group
    - AirConditioningFrontAndRear
    - AirConditioning
- Comfort
  - Optional
    - AutomaticHeadLights
- DrivingAndSafety
  - Optional
    - FrontFogLights
- Constraints
  - AutomaticHeadLights ⇒ FrontFogLights

**Hierarchy + Variability**  
= **set of valid configurations**

**configuration = set of features selected**

{CarEquipment, Comfort, DrivingAndSafety, Healthing, AirConditioning}

CarEquipment ✓

- Healthing ✓
  - Xor-Group
    - AirConditioningFrontAndRear ✗
    - AirConditioning ✓
- Comfort ✓
  - Optional
    - AutomaticHeadLights ✓
- DrivingAndSafety ✓
  - Optional
    - FrontFogLights ✗
- Constraints
  - AutomaticHeadLights ⇒ FrontFogLights

Optional    Xor-Group

Mandatory    Or-Group

**Hierarchy + Variability**  
= **set of valid configurations**

**configuration = set of features selected**

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

▼ CarEquipment

- ▼ ● Healthing
  - ▼ ▲ AirConditioningFrontAndRear
  - AirConditioning
- ▼ ● Comfort
  - AutomaticHeadLights
- ▼ ● DrivingAndSafety
  - FrontFogLights
- ▼ Constraints
  - AutomaticHeadLights ⇒ FrontFogLights

Optional ▲ Xor-Group

Mandatory ● Or-Group

## Hierarchy + Variability

=

## set of valid configurations

{CarEquipment, Comfort, DrivingAndSafety, Healthing}

⊗

{AirConditioning, FrontFogLights}

{AutomaticHeadLights, AirConditioning, FrontFogLights}

{AutomaticHeadLights, FrontFogLights, AirConditioningFrontAndRear}

{AirConditioningFrontAndRear}

{AirConditioning}

{AirConditioningFrontAndRear, FrontFogLights}

INFORMATIQUE ET SYSTEMES ALÉATOIRES

## Orthogonal Variability

CreditCard implies High

Specification in standardized CVL of base model variabilities

Variability Model

Base Model

Product Line model in any MOF compliant language

CVL execution

Product models fully described in the base language.

All regular base language tools can be applied to these models

Resolved Models

Resolution Models

Selection of a set of choices in the variability model

```

package com.alcatel;
import java.util.Date;
import java.util.Vector;

class Customer {
    private String name;
    private Vector<Rentals> _rentals = new Vector<Rentals>();

    public Customer(String name) {
        _name = name;
    }

    public String getMovie(Movie movie) {
        Rentals rental = new Rentals(movie, movie.getReleaseDate(), movie.getReleaseDate());
        _rentals.add(rental);
    }

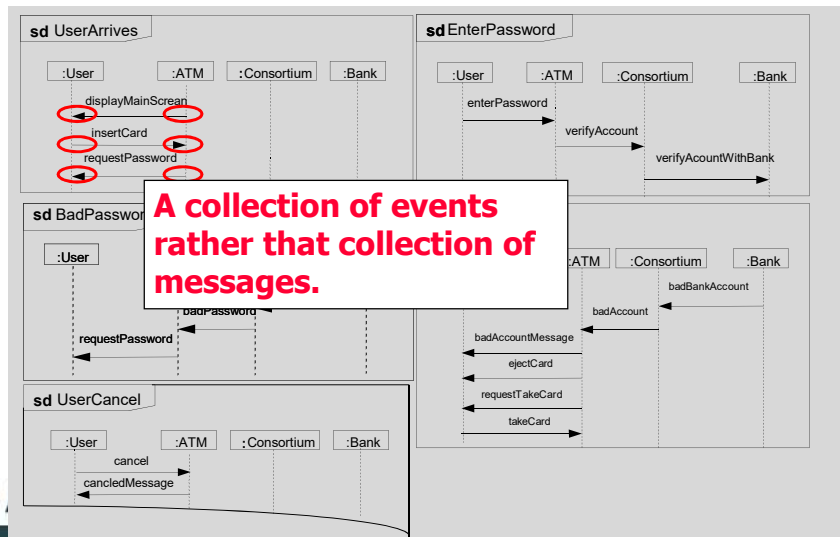
    public void addMovie(Movie movie) {
        _rentals.add(movie);
    }

    public String getMovie() {
        return _name;
    }
}
                    
```

INFORMATIQUE ET SYSTEMES ALÉATOIRES

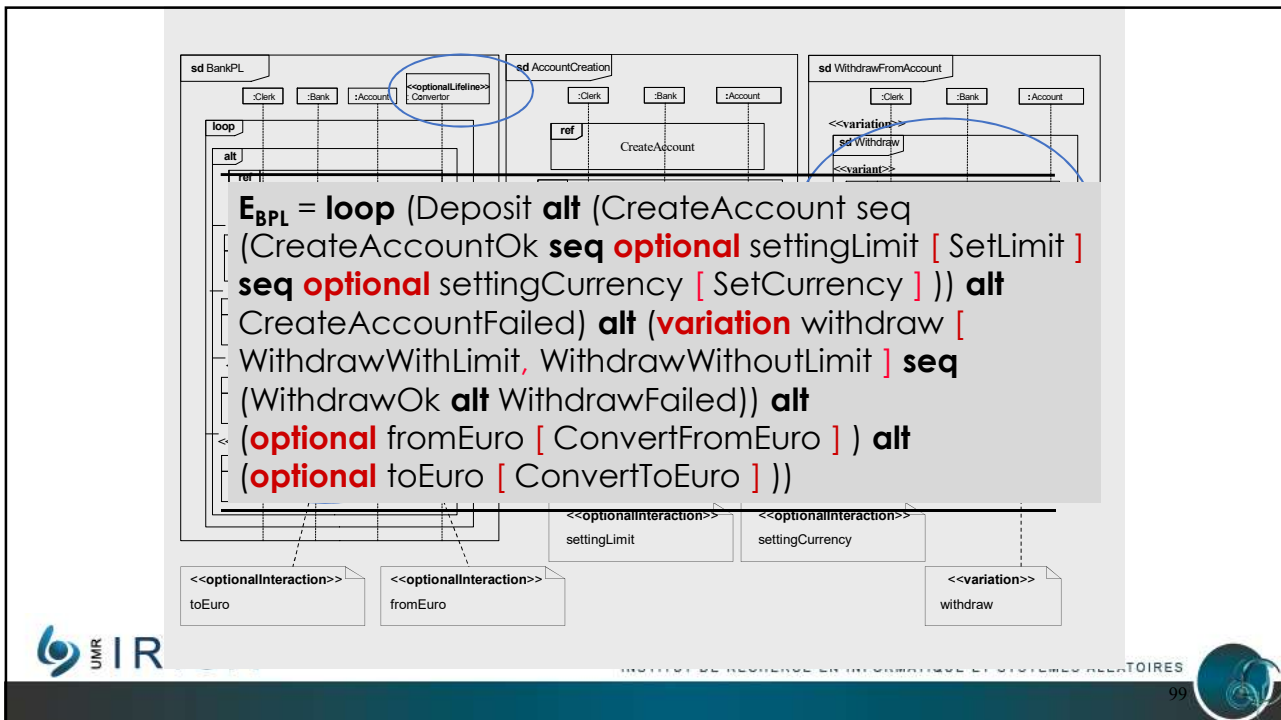


## Derivation of products from product-lines: behavioral issues based on Sequence Diagrams



## HMSCs & UML2.0 SDs: Combined SDs

- A combined SD: **refers** to a set of interactions and composes them by means of **operators**:
  - **Sequence (seq)**: weak sequential composition.
  - **Alternative (alt)**: choice between interaction operands.
  - **Loop (loop)**: iteration of an interaction.
- Extended with operators to model variability
  - **Optional, variation, virtual...**



## STEP 1: Behavioral derivation

Product	Decision model instance (DMI)
BS1	DM1 = {(settingLimit, TRUE), (settingCurrency, FALSE), (withdrawAccount, 1), (fromEuro, FALSE), (toEuro, FALSE)}
BS2	DM2 = {(settingLimit, FALSE), (settingCurrency, FALSE), (withdrawAccount, 2), (fromEuro, FALSE), (toEuro, FALSE)}
BS3	DM3 = {(settingLimit, FALSE), (settingCurrency, FALSE), (withdrawAccount, 2), (fromEuro, TRUE), (toEuro, TRUE)}
BS4	DM4 = {(settingLimit, TRUE), (settingCurrency, TRUE), (withdrawAccount, 1), (fromEuro, TRUE), (toEuro, TRUE)}

**RESD = [[PL-RESD]]<sub>DMi</sub>**

## Behavioral derivation rules

$$\bullet \text{ [[optional name [ E ] ] }_{DMi} = \begin{cases} E \text{ IF } (name, TRUE) \in DMi \\ E_{\emptyset} \text{ IF } (name, FALSE) \in DMi \end{cases}$$

$E_{\emptyset}$  is the *empty SD*: neutral element for **seq**, **alt** ; idempotent for **loop**

$$\bullet \text{ [[variation name [ E1, E2,.. ] ] }_{DMi} = \quad E_i \text{ IF } (name, i) \in DMi$$

$$\bullet \text{ [[virtual name [ E ] ] }_{DMi} = E_{\text{reff}} \text{ IF } (name, E_{\text{reff}}) \in DMi, E \text{ ELSE}$$



$DM2 = \{(settingLimit, FALSE), (settingCurrency, FALSE), (withdrawAccount, 2), (fromEuro, FALSE), (toEuro, FALSE)\}$

$E_{BS2} = [[ E_{BPL} ] ]_{DM2}$

$$E_{BS2} = \text{loop (Deposit alt (CreateAccount seq (CreateAccountOk seq } E_{\emptyset} \text{ seq (CreateAccountFailed) alt (WithdrawWithoutLimit seq (WithdrawOk alt WithdrawFailed)) alt ( } E_{\emptyset} \text{ ) alt ( } E_{\emptyset} \text{ ) ) )}$$

•  $E_{\emptyset}$  is the *empty SD*: neutral element for **seq**, **alt** ; idempotent for **loop**

→ expression reduction

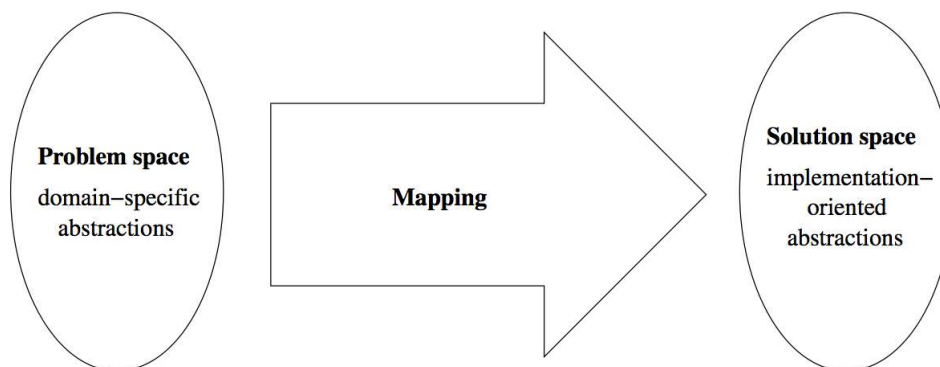


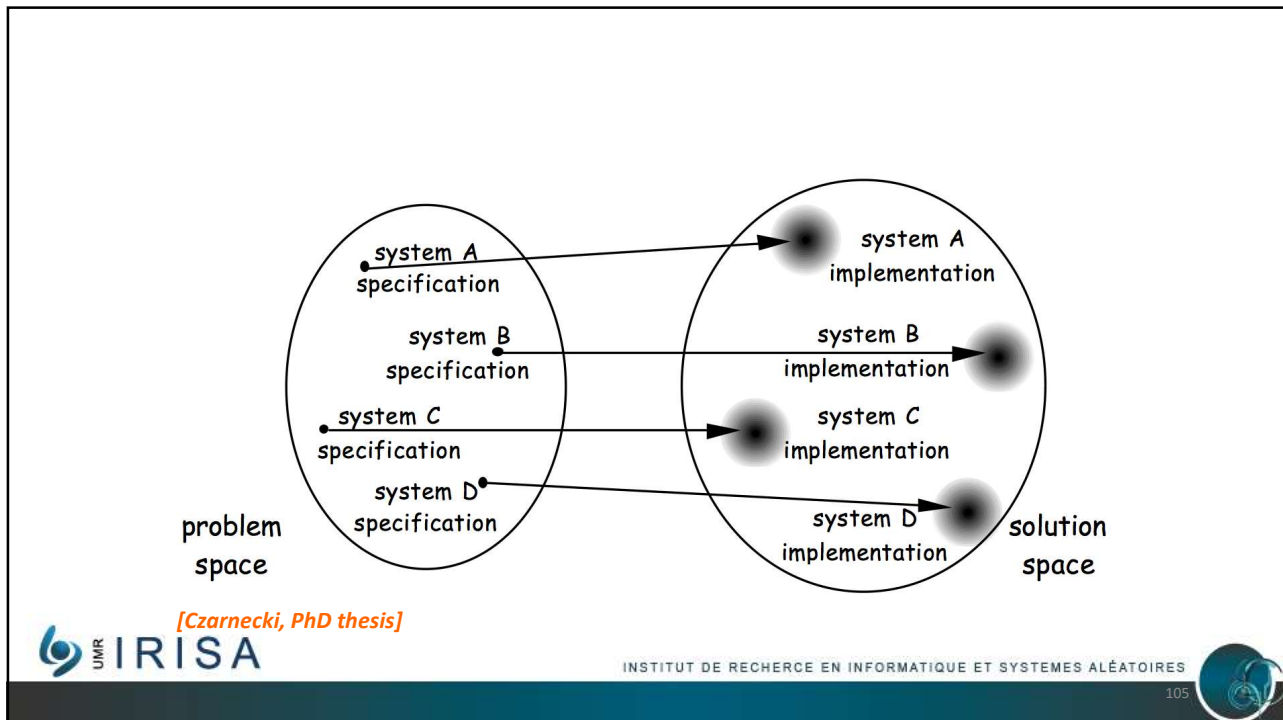
**DM2** = {(settingLimit, FALSE), (settingCurrency, FALSE), (withdrawAccount, 2),  
(fromEuro, FALSE), (toEuro, FALSE)}

$E_{BS2} = [[ E_{BPL} ]]_{DM2}$

$E_{BS2} = \text{loop}$  (Deposit **alt** (CreateAccount seq  
(CreateAccountOk )) **alt** CreateAccountFailed) **alt** (  
**WithdrawWithoutLimit** seq (WithdrawOk **alt**  
WithdrawFailed)))

**Result** : One expression (RES2) for each product





## Developing Product Lines

### Metamodels, DSLs, and Transformations to the rescue

- Domain Engineering
  - Domain Models
  - Level of abstraction
  - Domain-specific modeling languages
    - (visual or textual) syntacs, precise semantics
    - analyzed (verification)
  - Traceability between the artefacts
- Application Engineering
  - Model transformations (automation)
- Reduce the gap

# Summary

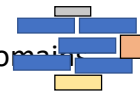
- **Software product line engineering**

- Mass customization
- Family of software intensive systems
- Systematic reuse
- Domain engineering
- Variability management



- **Variability everywhere**

- Applied and applicable to many industries and domains



- **Modeling and implementing variability: an overview**



## 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




## Acknowledgement



All these ideas have been developed with all my colleagues of the DiverSE team at IRISA/Inria

- *In particular M. Acher and O. Barais*

 @jmjezequel

[www.irisa.fr](http://www.irisa.fr)  @irisa\_lab

