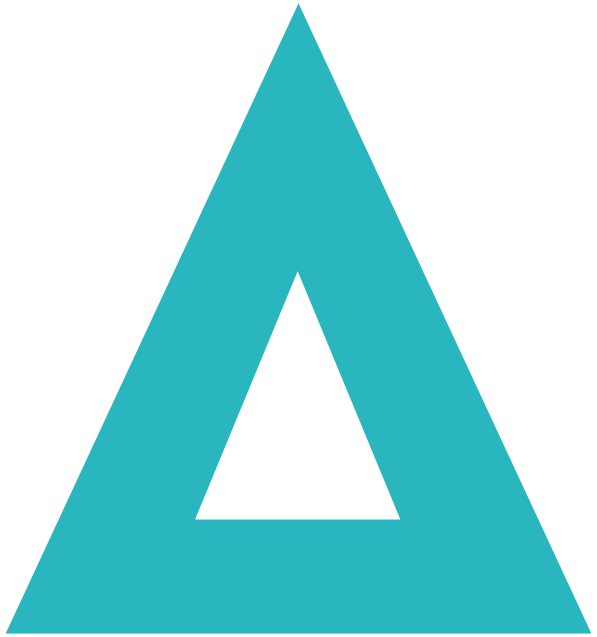
FLANDERS MAKE

FLANDERS MAKE

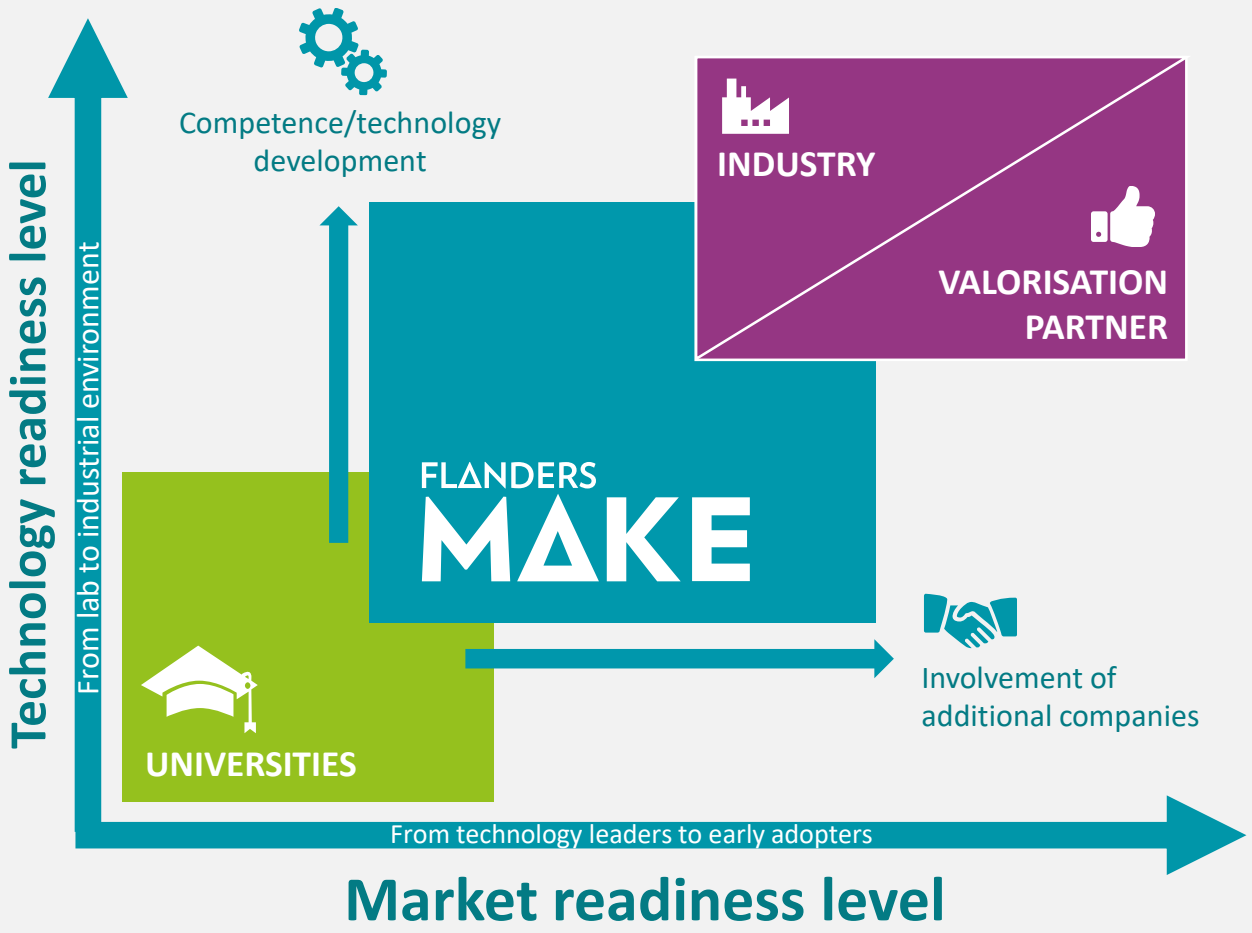
Role of Flanders Make



1. Look forward, follow trends and define research roadmaps
2. Anticipate, help formulate and tackle challenges and opportunities, to inspire companies
3. Strengthen companies' competences through a dynamic, collaborative innovation ecosystem
4. Create & transfer knowledge
5. Connect companies, research institutions and knowledge centres

FLANDERS MAKE

Innovation through collaboration



1 Universities lay the foundation, develop new ideas & technologies

2 Flanders Make involves partners and offers knowledge for further development

3 The product reaches technology and market readiness

FLANDERS MAKE MEMBER COMPANY ECOSYSTEM

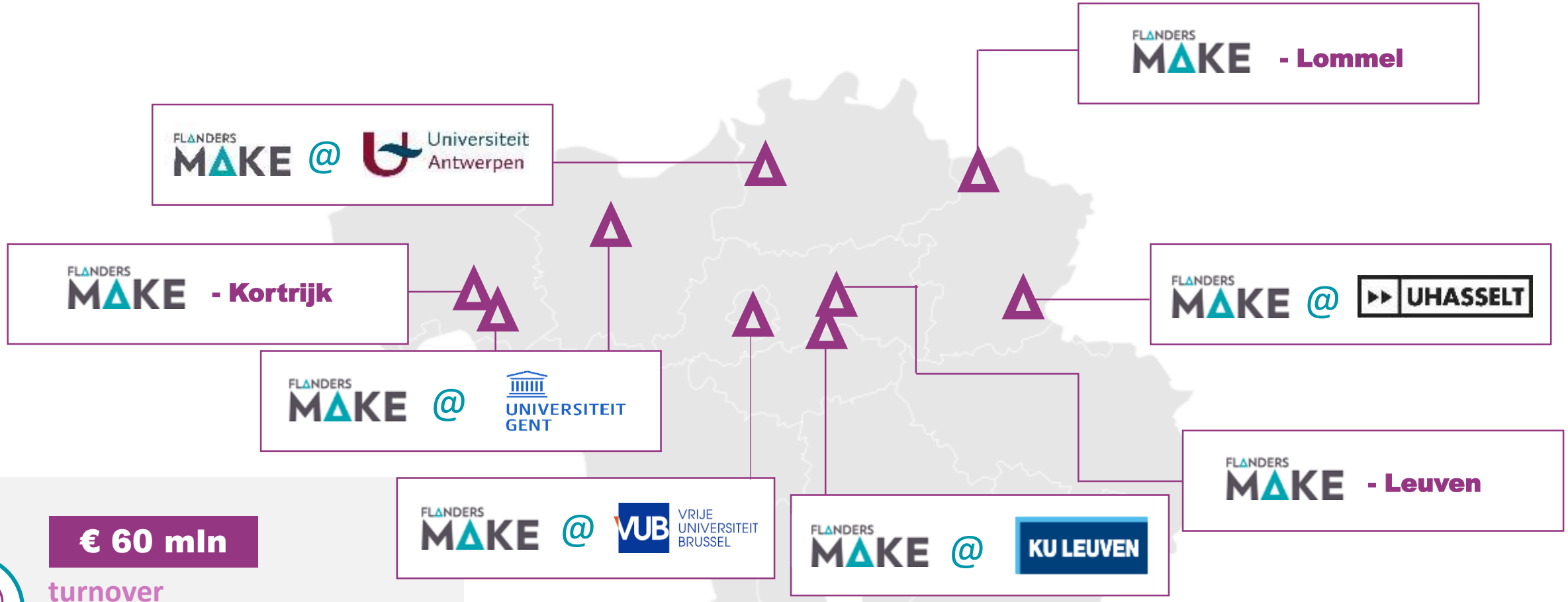
FLANDERS MAKE

2014: #56 ← **NUMBER OF MEMBERS** → **2019: #134**

Audi Brussels
Atlas Copco
Premium Sound Solutions
Bombardier
Signify
Continental
Janssen
Avery Dennison
Van Hool
Cegeka
Toyota
Bosch
Picanol Group
Punch Powertrain
Recticel
DAF
ABB
TE
CNH Industrial
CommScope
Niko
Rayvens
VCST
Dana
Barco
ZF
Daikin
Mariasteen
Tenneco
Bekaert
P&G
Deceuninck
Volvo
Volvo Trucks
ICO
Unilin
AGFA
VDL Bus & Coach
Siemens
LVD
Van de Wiele
asco
Terumo
Materialise
Soudal
Bosal
Grammer
Cronos
Zvhar
LayerWise
LommelProvingGround
Nikon
Voxdale
Knapp
ivex
CAWCO Technologies
Elgeterman
Optidrive
Pixelvision
Duco
Maintenance Partners
Azumuta
ESMA
Ninix Technologies
Jabil
televic rail
Defawes
u-sentric design, for people
Special machine construction Vaskon Mould manufacturing & repair
Vision++ Experts in Computer Vision
LCV Laser Cladding Venture
noesis Empower your experience
triphase
Arkite
Tremec
proceedix
Leuven air bearings
Powerdale
Supportsquare
Comate
Pearl Chain
Melotte
Xenics Infrared Solutions
Trucks Europe
vitalo
oerlikon
FRS
Dentsply Implants
Arcomet
Transport & Mobility Leuven
achillesdesign
Octinion
Provan
Maex
Alberts
Adcax
CadCorner
BO Enterprises
AMS Belgium
Rhea
Waak Mensgericht Ondernemen
Flagstone
kobi
Dynamic Dimensions
FiXar
Lambrecht
Averna
.Ocean nautical innovations
LVA
FLIR
SL Sabca Limburg NV
Laborelec GDF Suez
Kerv
Elnor Motors
exmore making industrial standards work A KOMAX COMPANY
MLG
Cocas
Hela
yazzoom creating value from data
INDIGI
Inverto
Magnax
Objective Supply chain execution software
Peceo

FLANDERS MAKE

3 co-creation centres and labs at the 5 Flemish Universities



€ 60 mln

turnover

600

highly specialised researchers

FLANDERS MAKE

3 co-creation centres

Co-creation Centre

Vehicle development

Lommel

4200 m²

Co-creation Centre

Machine development

Leuven

3500 m²

Co-creation Centre

Customised production - Industry 4.0

Kortrijk

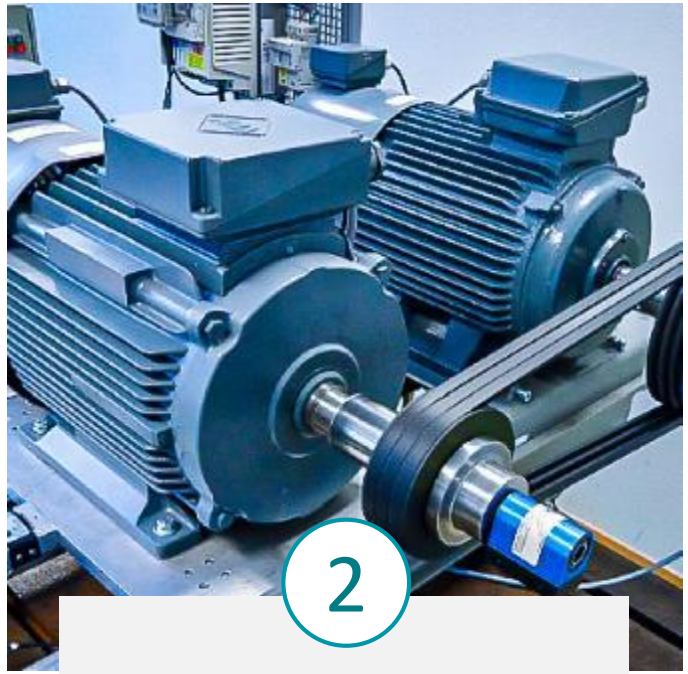
6000 m²

FLANDERS MAKE

Fields of application

1

Production



2

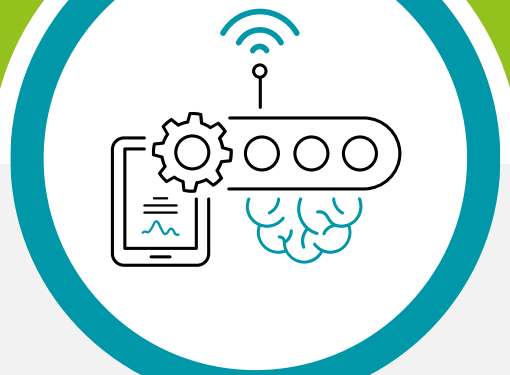
Machines

3

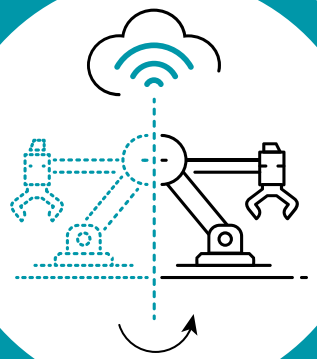
Vehicles

FLANDERS MAKE

Research supporting the digitalisation of the manufacturing industry clustered around 4 competences

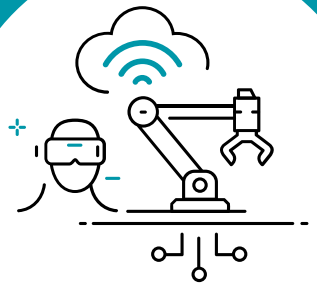


Decision & Control



Design & optimisation

Motion products



Flexible assembly

FLANDERS MAKE

Design & optimisation

Research track

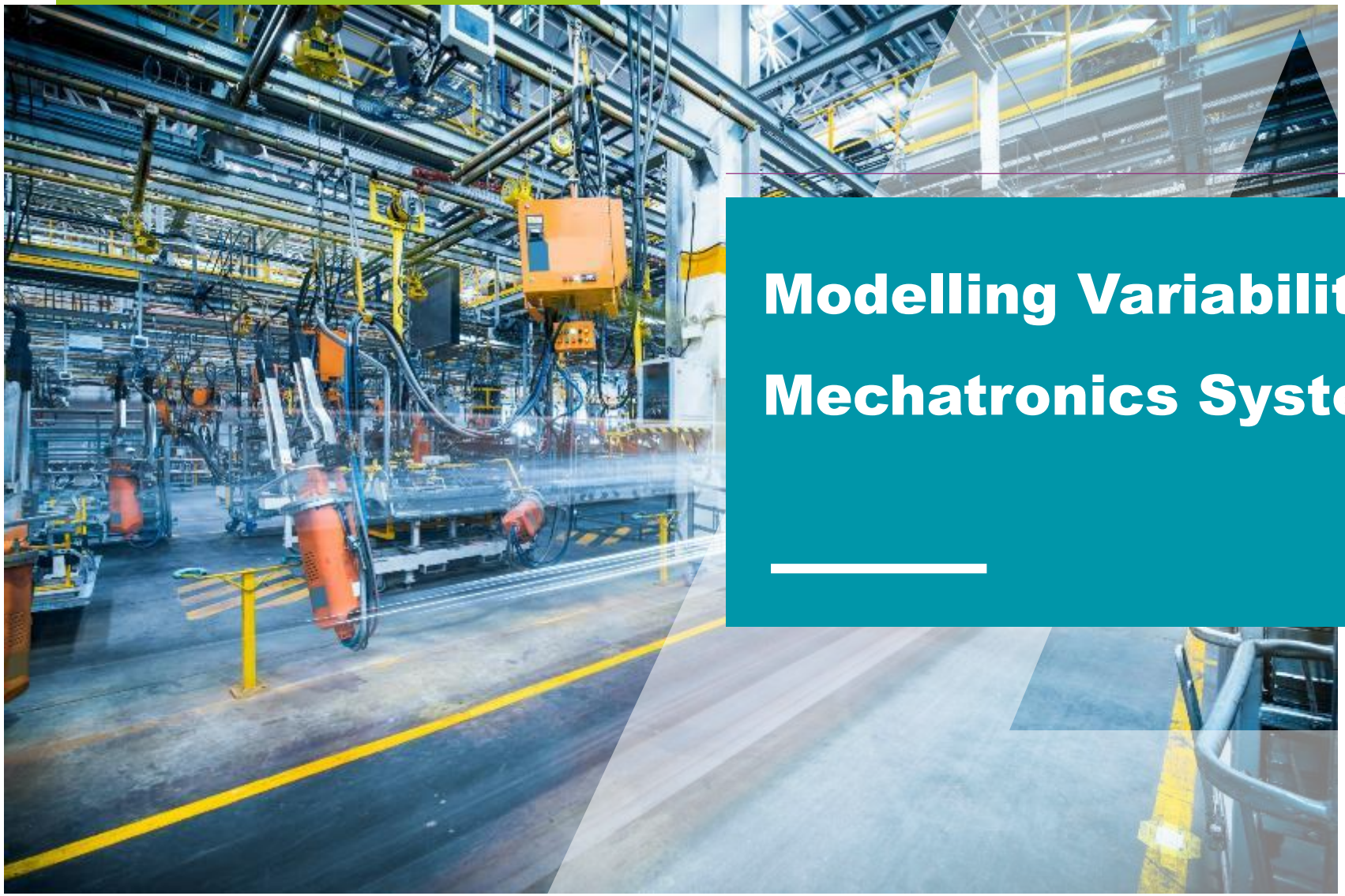
Design & optimisation

GOAL

Methods and tools to manage design complexity & efficiency

HOW AND WHICH TECHNOLOGIES?

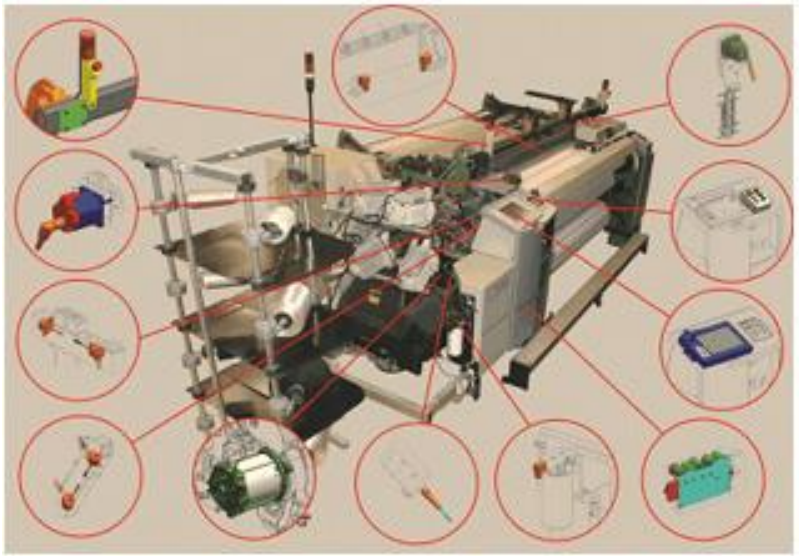
1. **Simultaneous** model-based **design of hardware & software** for smart systems
2. Model-based design of **assembly systems enabling high product variability**
3. Design that takes into account the (financial) **impact of product lifecycle phases**
4. Efficient design of **product variants & product families**



Modelling Variability in Mechatronics Systems Design

Smart, interconnected products and production systems

Mechanical system

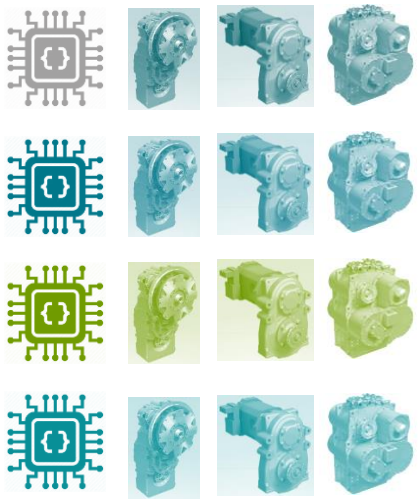


Mechatronic system

Daimler (press release 2015)
Mercedes-Benz E-class
100 mio lines of code

Increase in components, functions, technologies & disciplines **=> higher COMPLEXITY**

Customized products



One product

Product family

Mass customisation

Increase in variants => **higher COMPLEXITY**

Managing Variability

https://www.youtube.com/watch?v=nd4tPauouQk

DSM-TP 2016

Modeling Variability

Andrzej Wąsowski

VARIETE, DFF SAPERE AUDE

VARIES

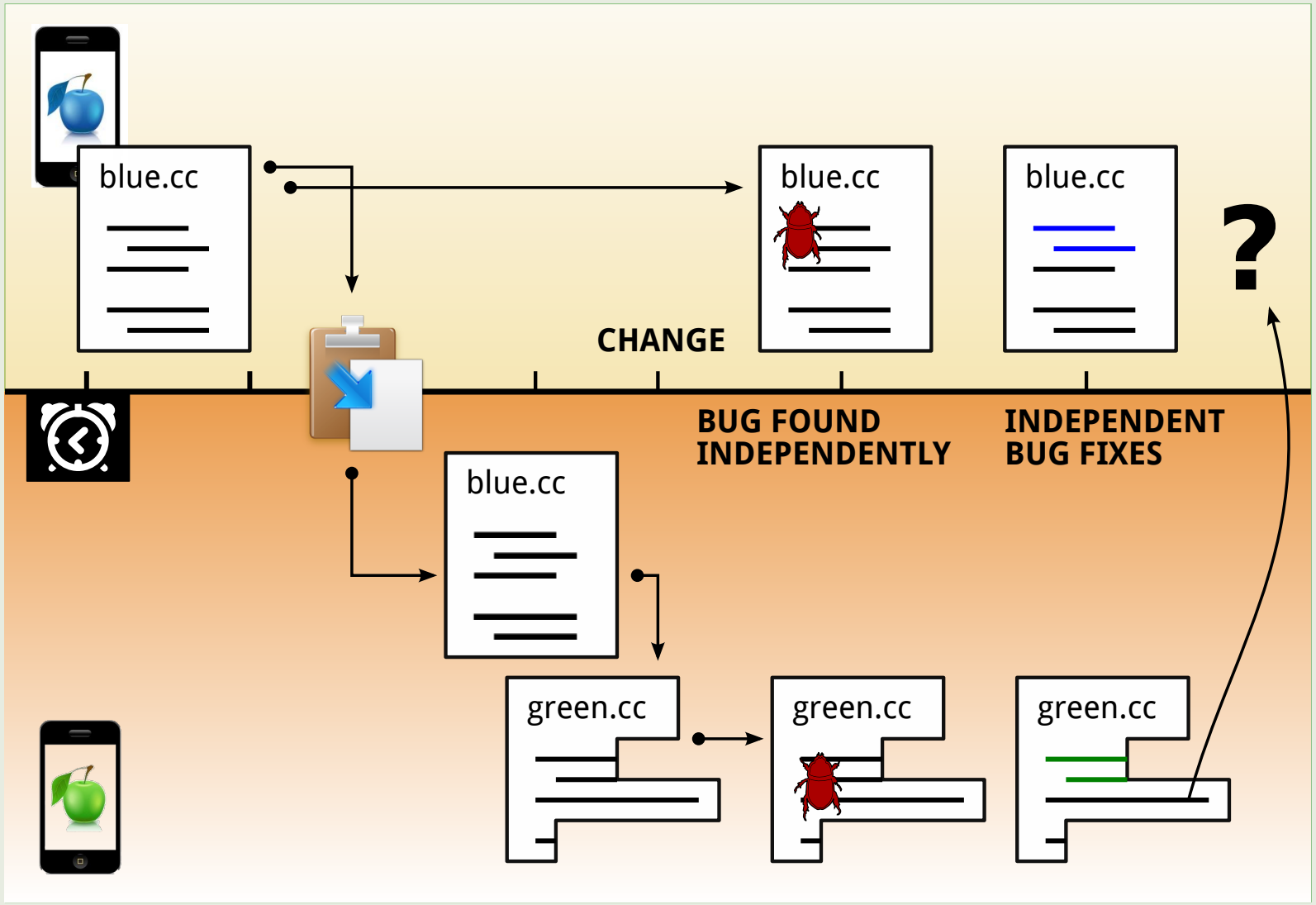
cost

MPM4CPS

IT UNIVERSITY OF COPENHAGEN

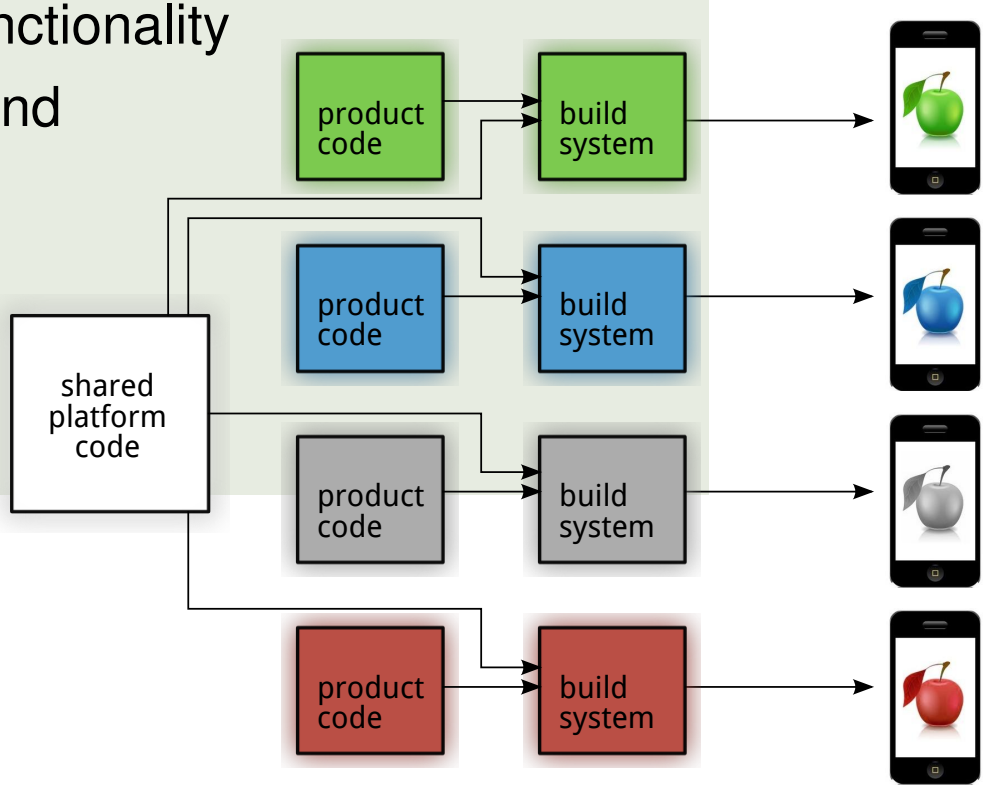
PROCESS AND SYSTEM MODELS GROUP

Drowning in Clone-And-Own



Opportunistic Reuse Does Not Work

- Common scenario:
  - version the code, reuse when **opportunity** appears
- If the file to be reused needs change, **copy** it
  - You *clone-and-own* it
- **Benefit** from **quickly** available functionality
- But have to test, debug, change and evolve the file **yourself**
- **Product specific** code grows
- Platform code **diminishes and degrades**



SUCCESSFUL REUSE IS

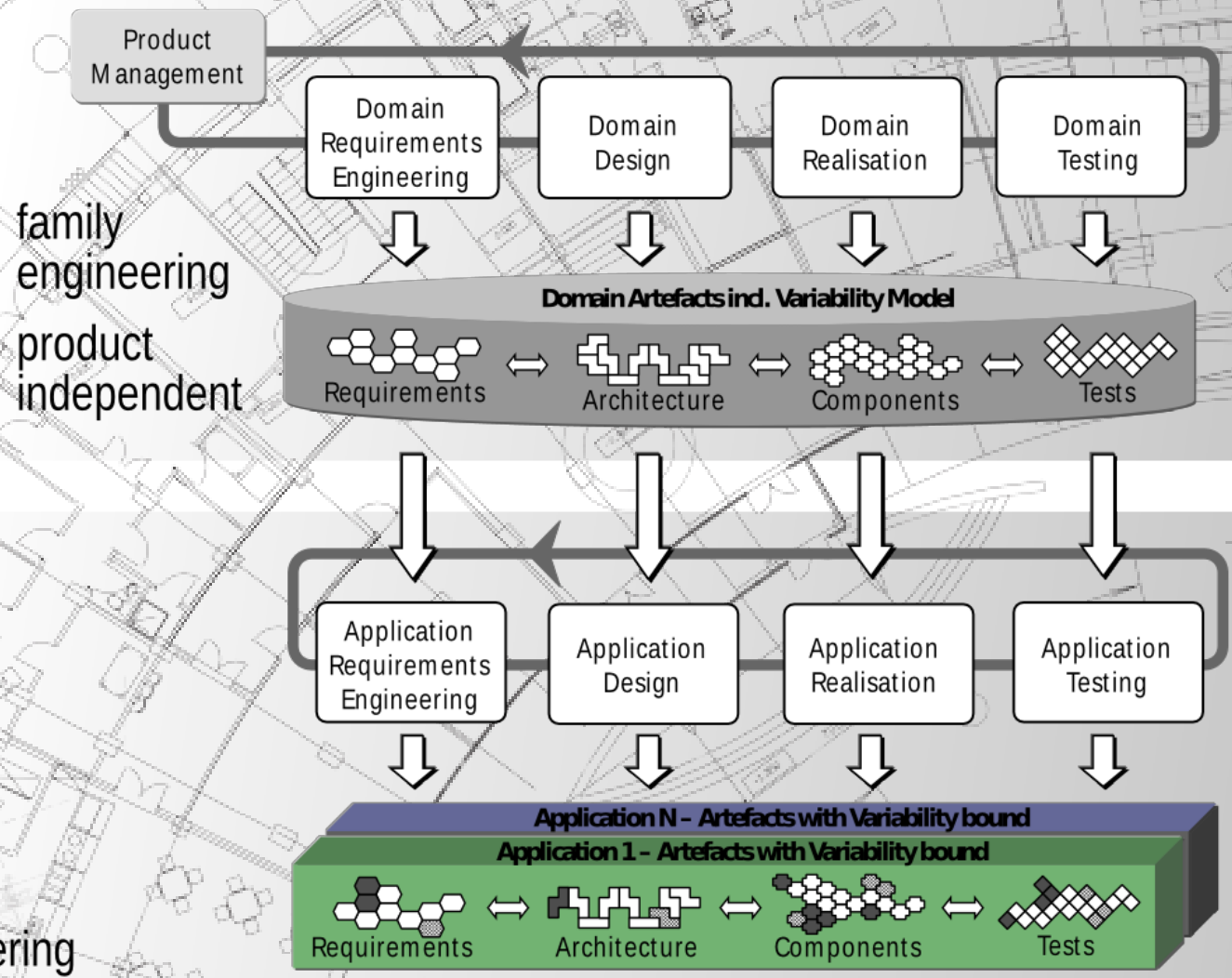
PROACTIVE
PLANNED
MANAGED

- SPL Method, **Architecture**
- Variability **Implementation Spectrum**
- Variability Abstraction: **Feature Modeling**
- Variability Modeling in **Practice**
- Variability **Realization**

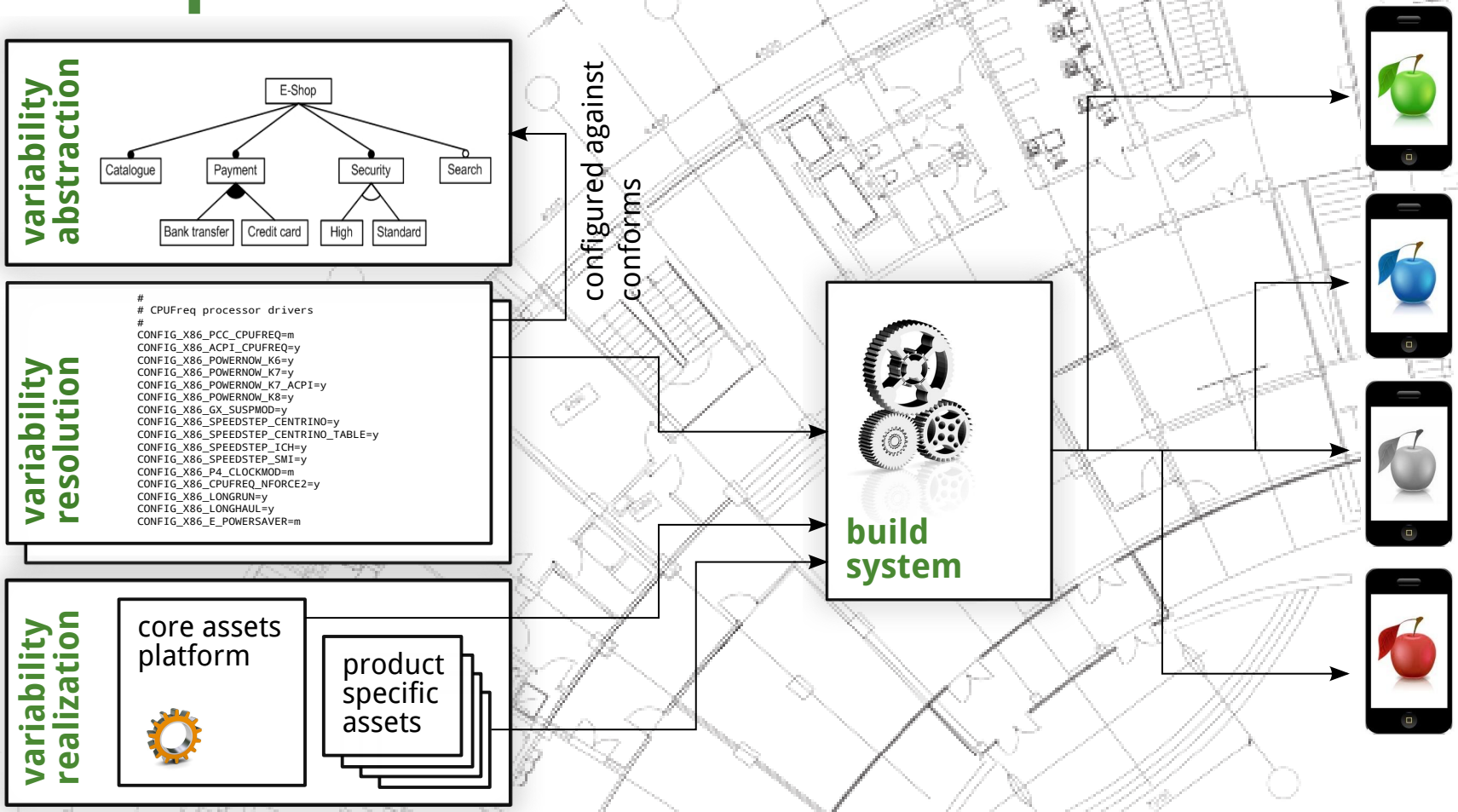
AGENDA

Domain vs Application Processes

Pohl et. al. Software Product Line Engineering



A Simple Product Line Architecture



- Less product specific = more reuse: development/tests/debugging/build
- Model of *commonality* and *variability*.
- Scope under control. Explicit feature life cycle

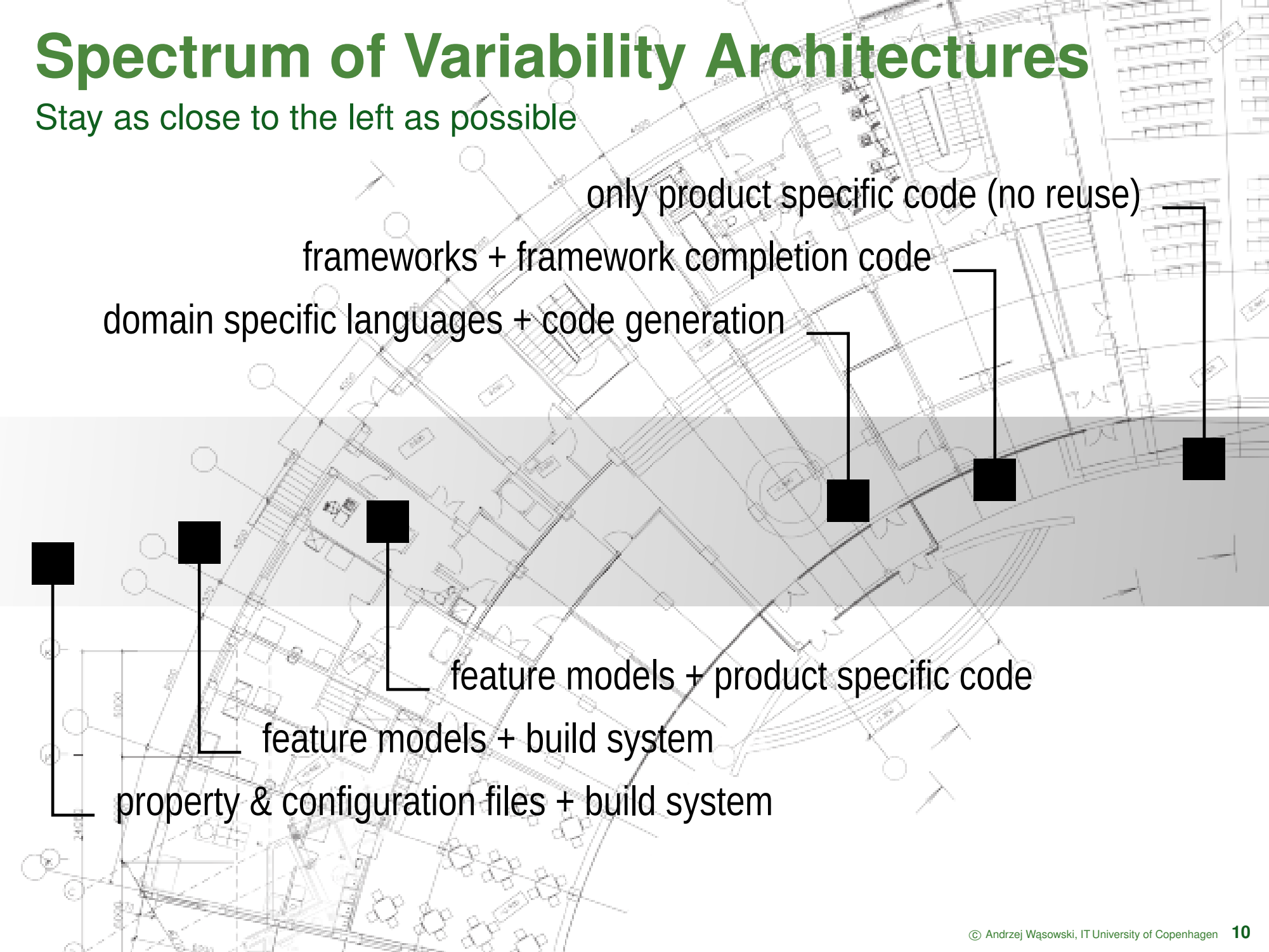
Exploit commonality
Manage variability

- SPL Method, **Architecture**
- Variability **Implementation Spectrum**
- Variability Abstraction: **Feature Modeling**
- Variability Modeling in **Practice**
- Variability **Realization**

AGENDA

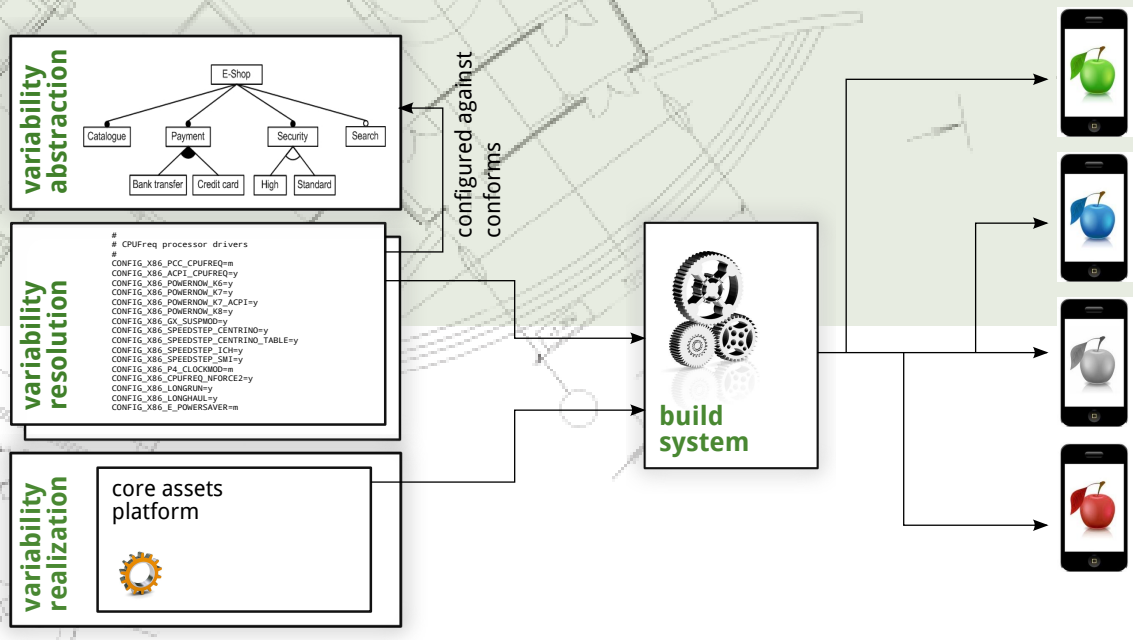
Spectrum of Variability Architectures

Stay as close to the left as possible



Implementation Technologies

- Variability abstraction: FMs, DSLs, or **none**
- Variability resolution:
  - XML property file
  - FM configuration
  - Domain specific model (DSM)
- Variability realization:
  - general purpose code
  - w/ variability techniques
  - code generators
  - model transformers
  - parts may use DSLs
  - etc.



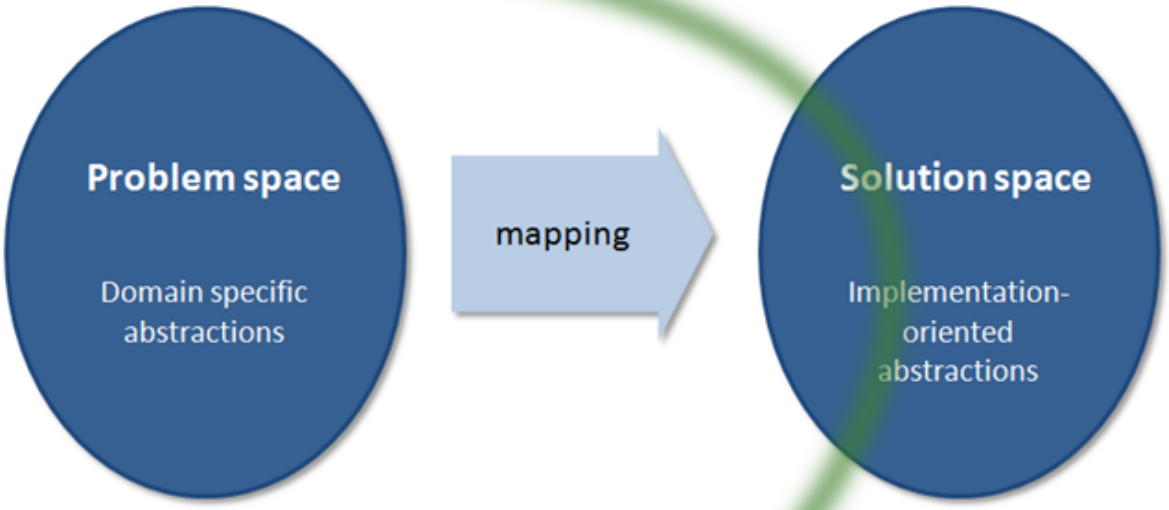
- SPL Method, **Architecture**
- Variability **Implementation Spectrum**
- Variability Abstraction: **Feature Modeling**
- Variability Modeling in **Practice**
- Variability **Realization**

AGENDA

Problem Space

Solution Space

features



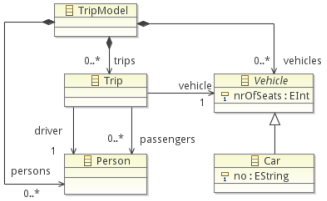
variation points

SMS notification on transaction

- paid services

- SMSLoggerAspect
- Phone No. in data model

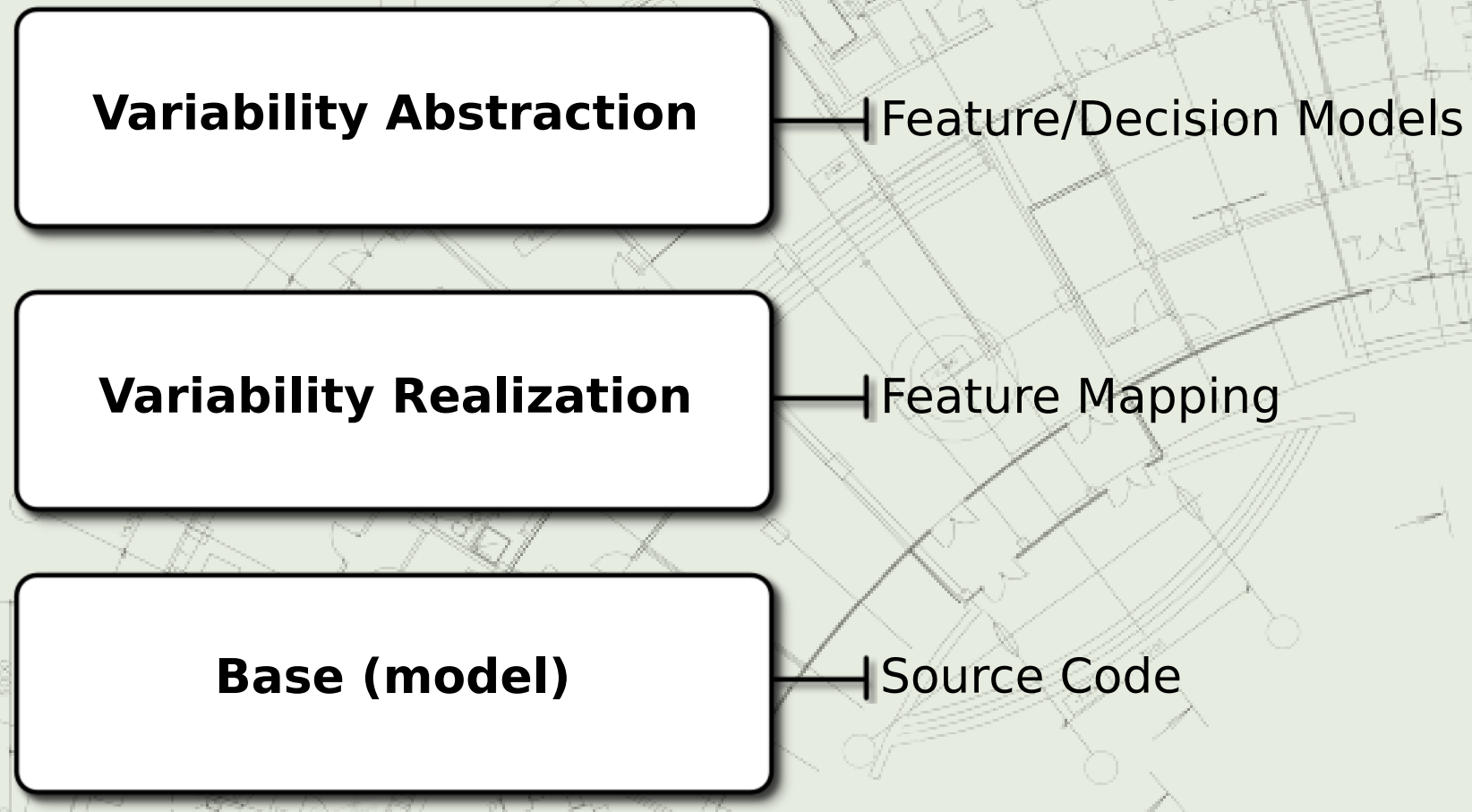
invoice



code.cs

CVL Architecture for Dummies

The degree of coupling can be controlled by moving the mapping



CVL submitters. *Common Variability Language, OMG Revised Submission*. 2012

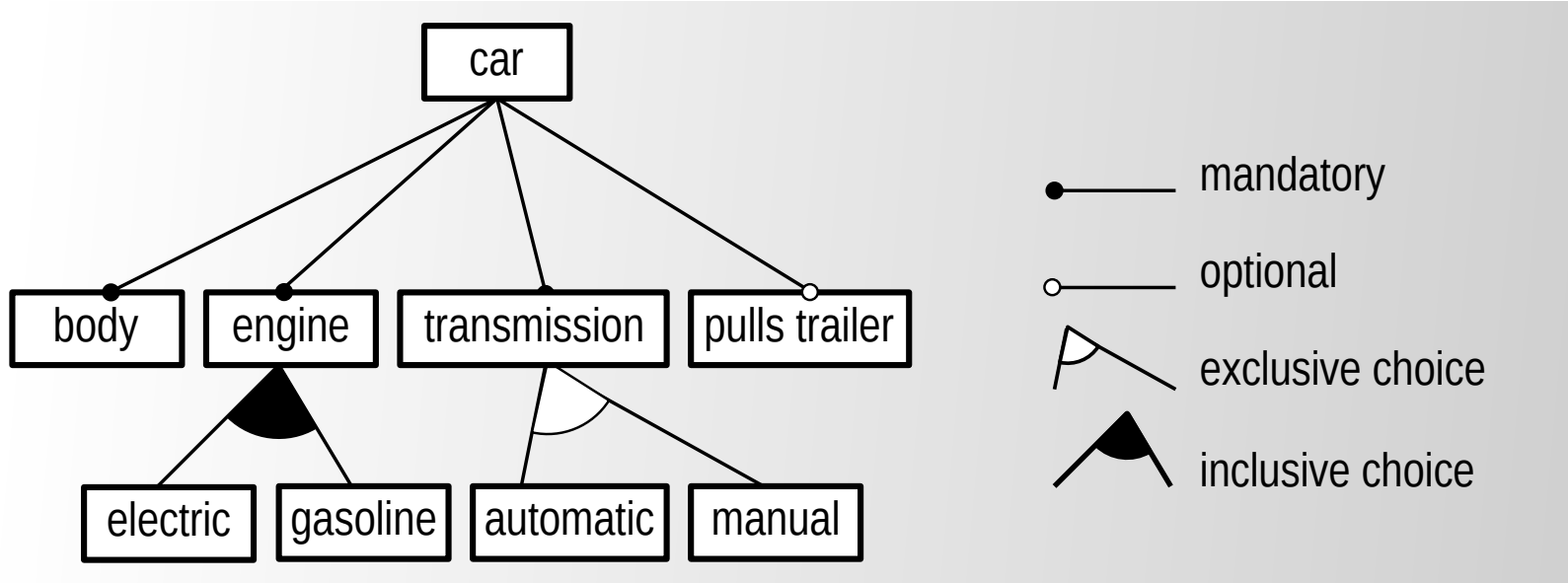
Feature Modeling (I)

feature:
a single variability increment in the problem domain (decision)

variation point:
a single variability increment in the solution space

Feature Modeling (II)

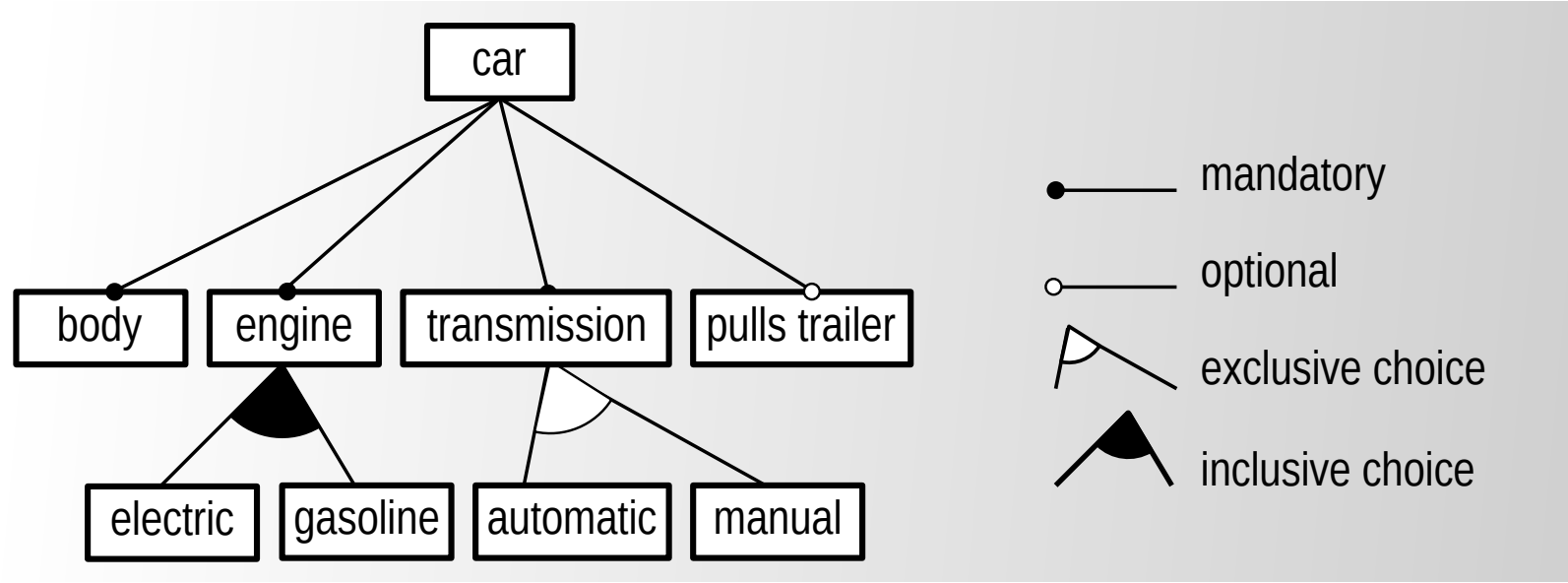
Example from Czarnecki'02



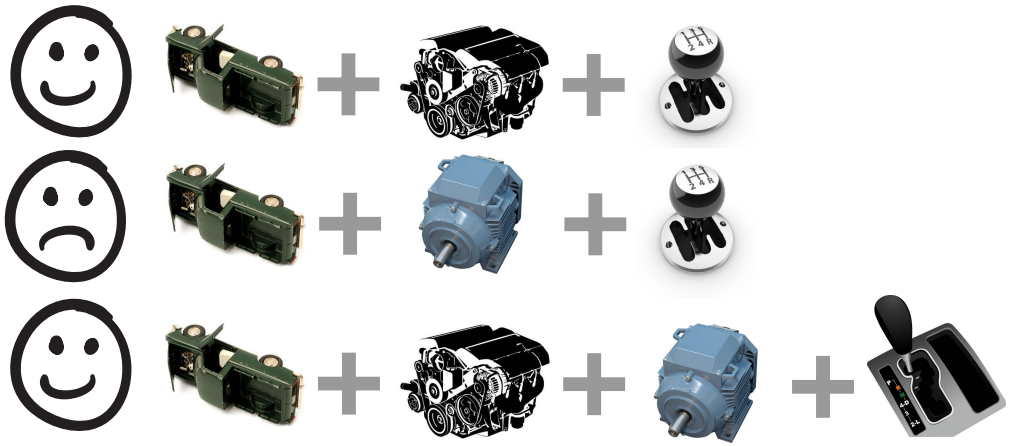
- **Hierarchy** constraints, for example:
  - manual requires transmission (each child node requires its parent node)
- **Groups** constraints: engine is electric or gas driven or both
- Not all constraints in hierarchy & groups, cross-tree constraints in text:
  - electric *requires* automatic
- Attributes are added like to classes (eg. engine volume)

Feature Modeling (III)

Configuration

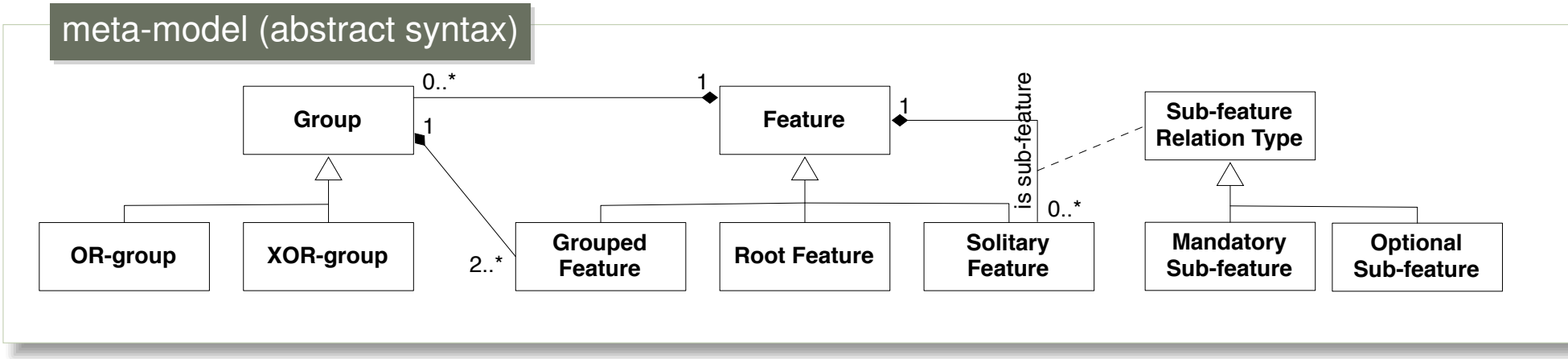


electric *requires* automatic



Feature Models (IV)

An example meta-model from Janota'08

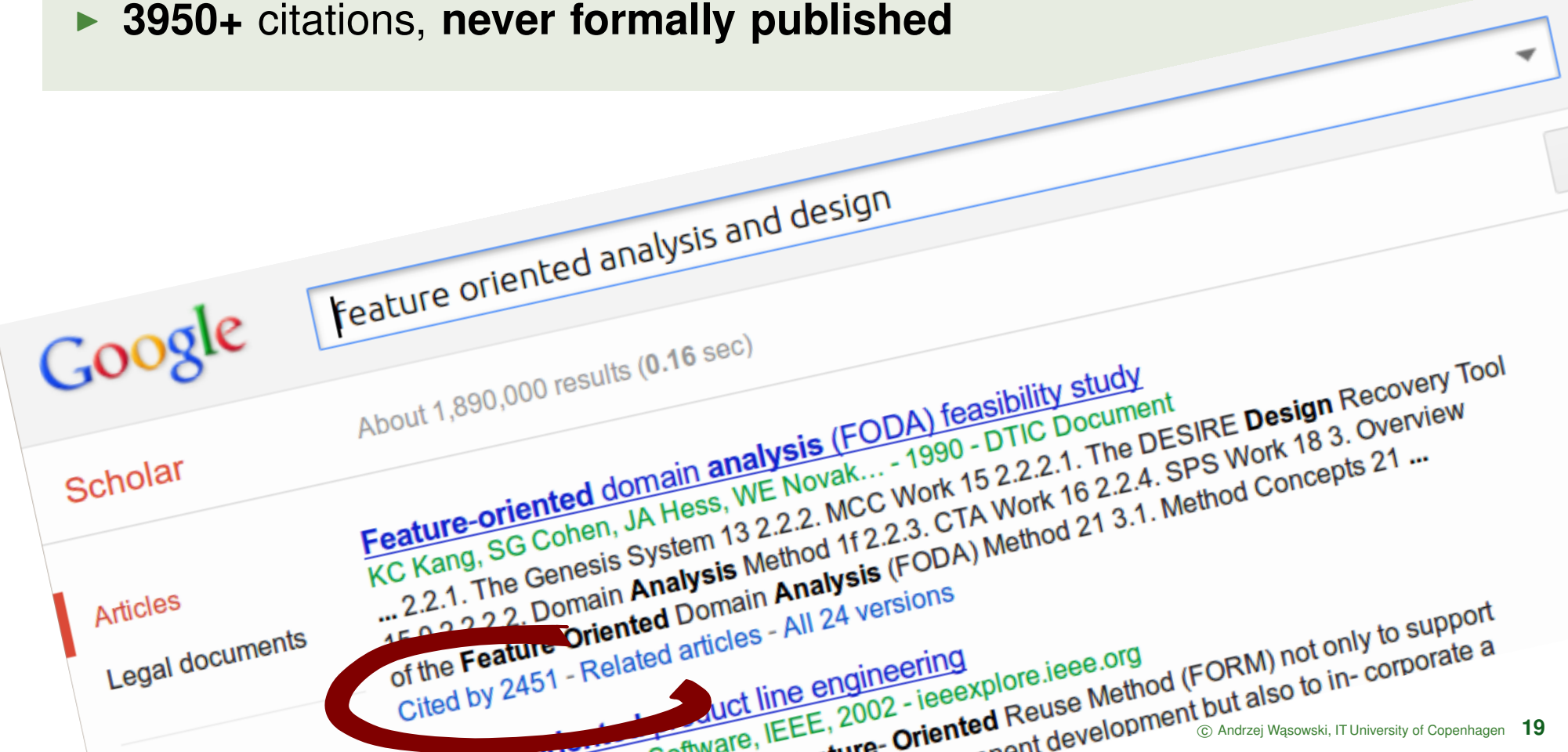


- Note a single generic kind of relations: **subfeature**
- No distinction between **kind-of** (inheritance) and **part-of** (containment), like class modeling does
- A characteristic feature of **configuration and constraint languages** (as opposed to structural modeling languages)
- Clafer (as a structural modeling langauge) supports the distinction, but so do other feature modeling languages

Feature Modeling and FODA

Feature Oriented Design and Analysis by Kang et al. 1990

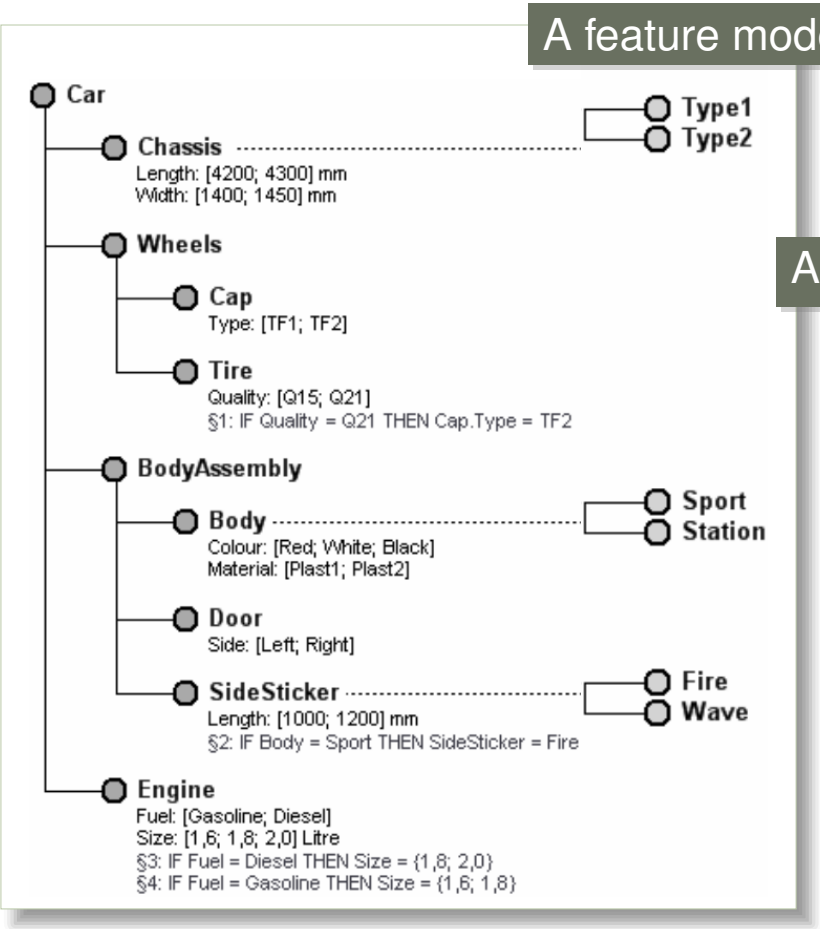
- **FODA succeeds for its simplicity**
- Probably best intro in Czarnecki's *Generative Programming* (Chpt. 4)
- **3950+** citations, **never formally published**



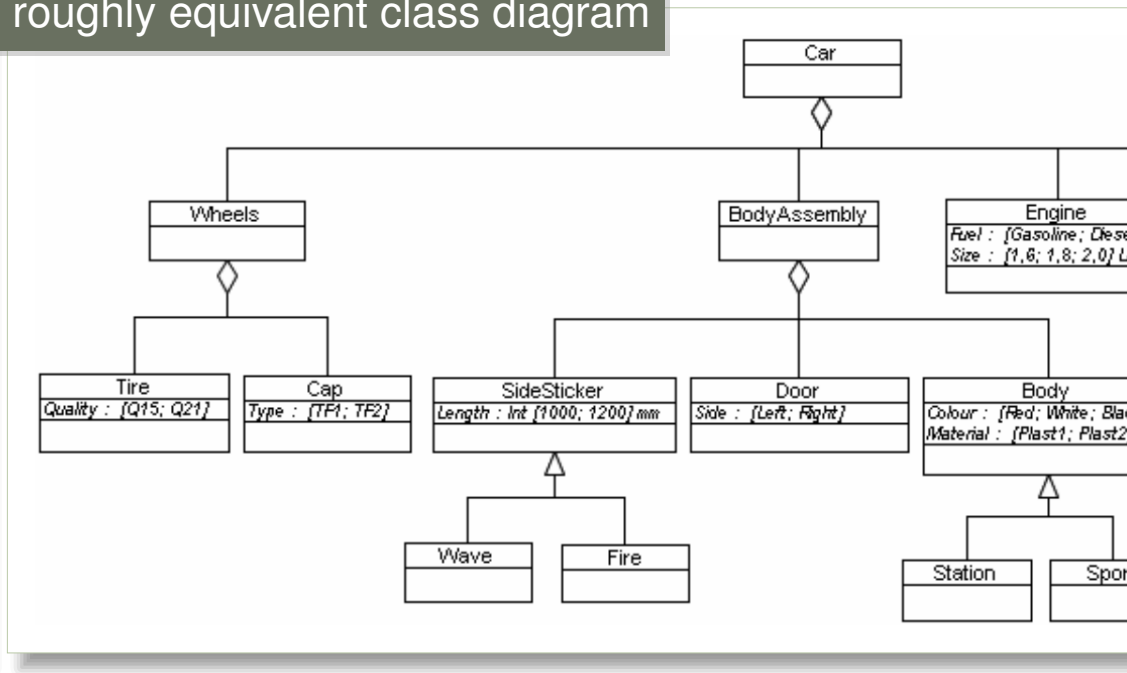
Feature Modeling vs Class Modeling

A feature model in *Product Variant Master* Notation (Hvam)

A feature model



A roughly equivalent class diagram

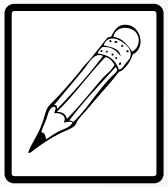


More on this: Bąk. Czarnecki. Wąsowski. *Feature and Meta-Models in Clafer: Mixed, Specialized, and Coupled*. SLE 2010

Above models from: Haug. Degn. Poulsen. Hvam. *Creating a documentation system to support the development and maintenance of product configuration systems*

Applications of Feature Models

Design & Management

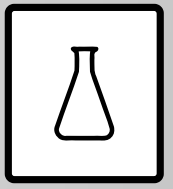


domain
modeling

product line scoping
product line mngmt

code generation
driving build system

driving
testing



Development & Test

How To Build Feature Models?

Two strategies, but only one good :)

top-down

- **Big-bang** adoption
- Perform **careful domain analysis**
- Document **concepts, abstractions and relations** between them in a FM

bottom-up

- Identify a **cloned component**
- Find the **patches** that describe differences
- Translate **diffs to variation points**
- Organize **variation points into features**, and a hierarchy
- Works well with **incremental adoption**
- See SPLC07 paper by Danfoss

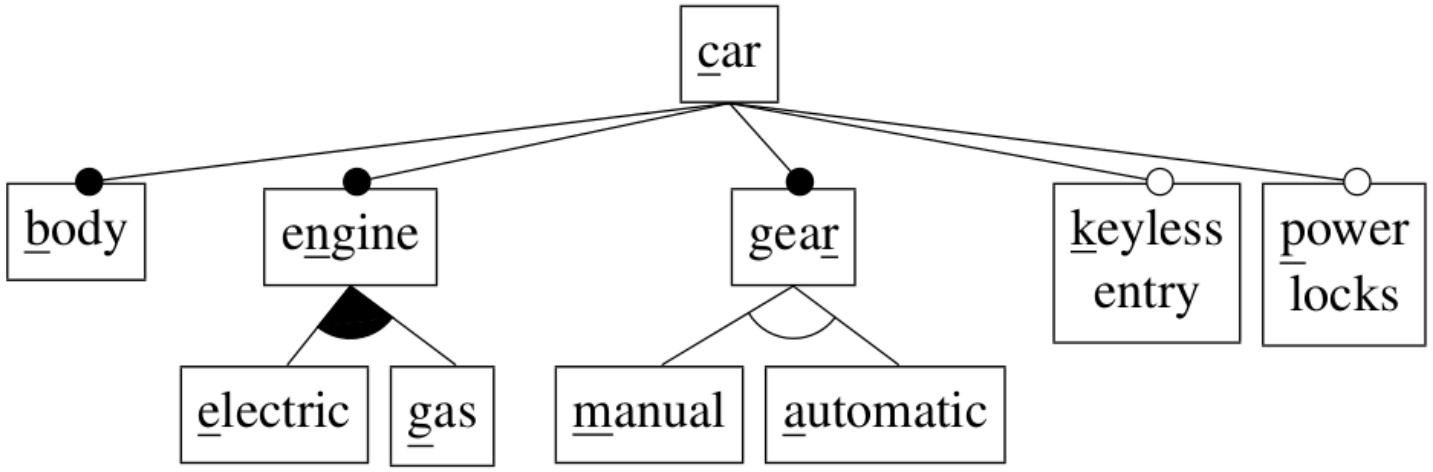
Hans Peter Jepsen, Jan Gaardsted Dall, Danilo Beuche. *Minimally Invasive Migration to Software Product Lines.* SPLC 2007

- SPL Method, **Architecture**
- Variability **Implementation Spectrum**
- Variability Abstraction: **Feature Modeling**
- Variability Modeling in **Practice**
- Variability **Realization**

AGENDA

Variability Modeling is The Success Story of Modeling

A Laboratory Feature Model

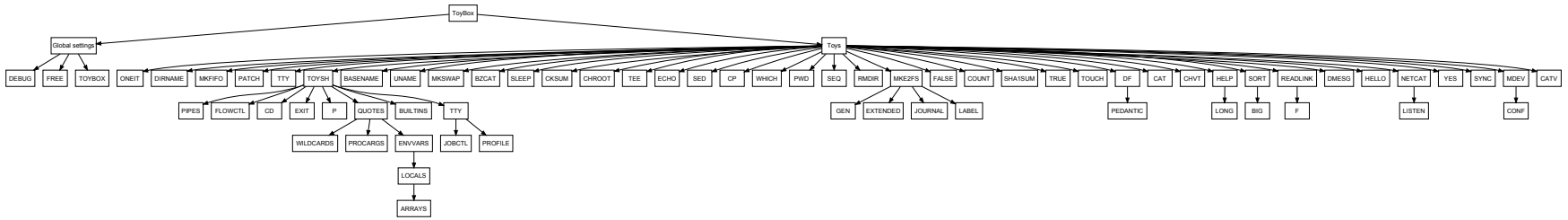


$$keyless\_entry \rightarrow power\_locks$$

Czarnecki, Wąsowski. *Feature Diagrams and Logics: There and Back Again.* In: 11th International Software Product Line Conference (SPLC 2007) Kyoto, Japan, 10-14 September, 2007 © IEEE Press.

A Healthy Wild Feature Model Cub

ToyBox project, 71 features



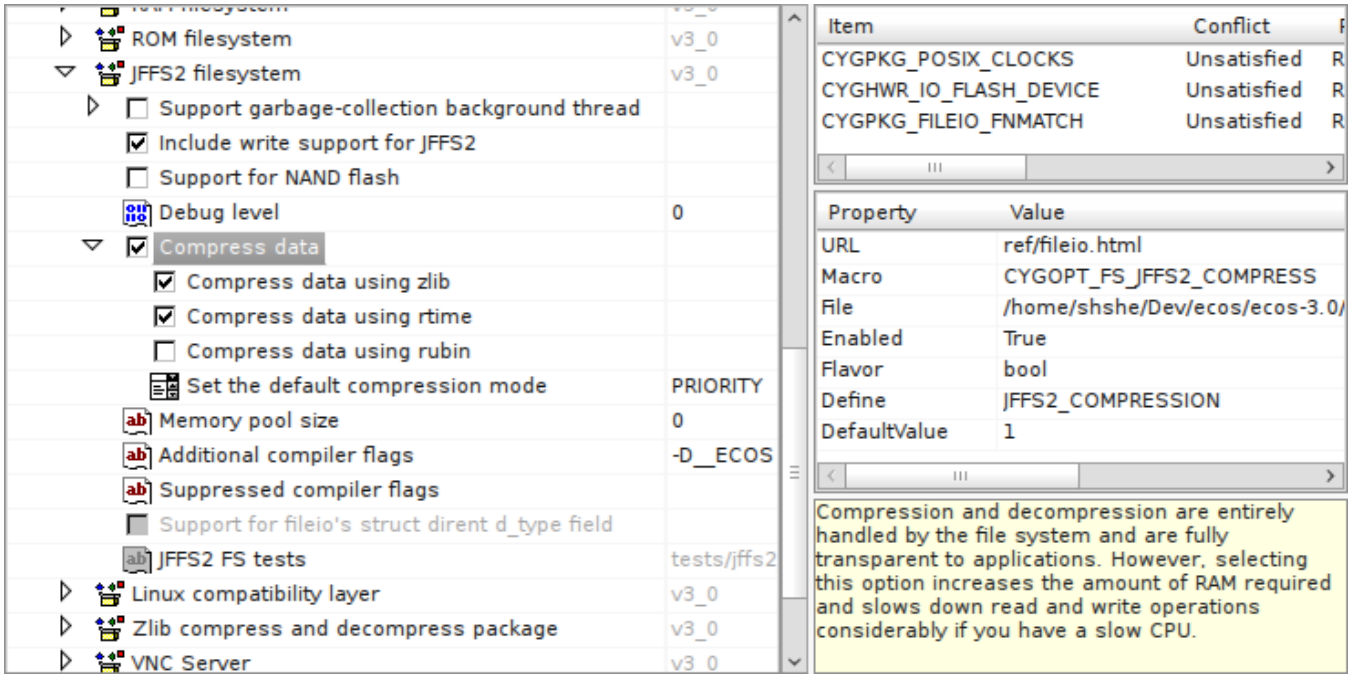
The Linux Kernel has 6-12K features, depending how you count!
But maximum depth is 8, most leaves are at 4!

↓ this is the Linux kernel model fit to the slide width ↓

Berger, She, Lotufo, Wąsowski, Czarnecki. *A Study of Variability Models and Languages in the Systems Software Domain.* IEEE Transactions in Software Engineering, 2013

Is FODA special? Not really!

eCos configurator



- **Linux kernel** and **eCos operating system** use similar configurator, controlled by **textual variability models**
- **Trees become unwieldy** very fast
- Many tools used **linearized trees**, like above
- Nice **trees are good for PowerPoint**, whiteboard and brainstorming

It is easy to design a textual syntax

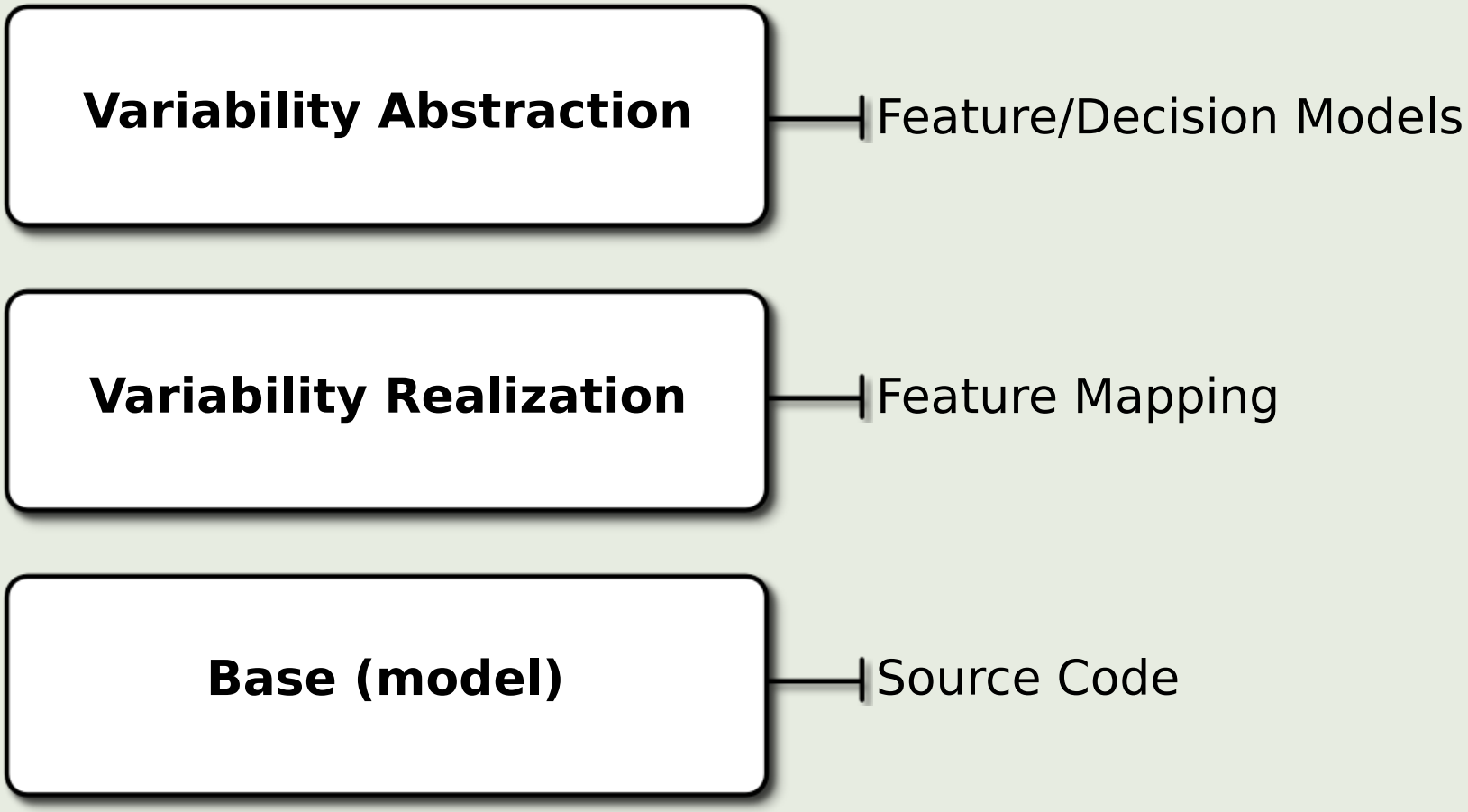
Kconfig of Linux kernel, designed by non-experts, with no tools
CDL of eCos, designed by non-experts, with no tools

```
k-1   menuconfig MISC_FILESYSTEMS
k-2     bool "Miscellaneous filesystems"
k-3
k-4    if MISC_FILESYSTEMS
k-5
k-6     config JFFS2_FS
k-7      tristate "Journalling Flash File System" if MTD
k-8      select CRC32 if MTD
k-9
k-10
k-11
k-12
k-13    config JFFS2_FS_DEBUG
k-14     int "JFFS2 Debug level (0=quiet, 2=noisy)"
k-15     depends on JFFS2_FS
k-16     default 0
k-17     range 0 2
k-18     --- help ---
k-19       Debug verbosity of ...
k-20
k-21
k-22    config JFFS2_FS_WRITEBUFFER
k-23     bool
k-24     depends on JFFS2_FS
k-25     default HAS_IOMEM
k-26
```

```
c-1   cdl_component MISC_FILESYSTEMS {
c-2    display   "Miscellaneous filesystems"
c-3    flavor    none
c-4   }
c-5
c-6   cdl_package CYGPKG_FS_JFFS2 {
c-7     display     "Journalling Flash File System"
c-8     requires    CYGPKG_CRC
c-9     implements  CYGINT_IO_FILEIO
c-10    parent      MISC_FILESYSTEMS
c-11    active_if   MTD
c-12
c-13    cdl_option CYGOPT_FS_JFFS2_DEBUG {
c-14     display         "Debug level"
c-15     flavor          data
c-16     default_value   0
c-17     legal_values    0 to 2
c-18     define          CONFIG_JFFS2_FS_DEBUG
c-19     description     "Debug verbosity of...."
c-20    }
c-21
c-22    cdl_option CYGOPT_FS_JFFS2_NAND {
c-23     flavor          bool
c-24     define          CONFIG_JFFS2_FS_WRITEBUFFER
c-25     calculated      HAS_IOMEM
c-26    }
```

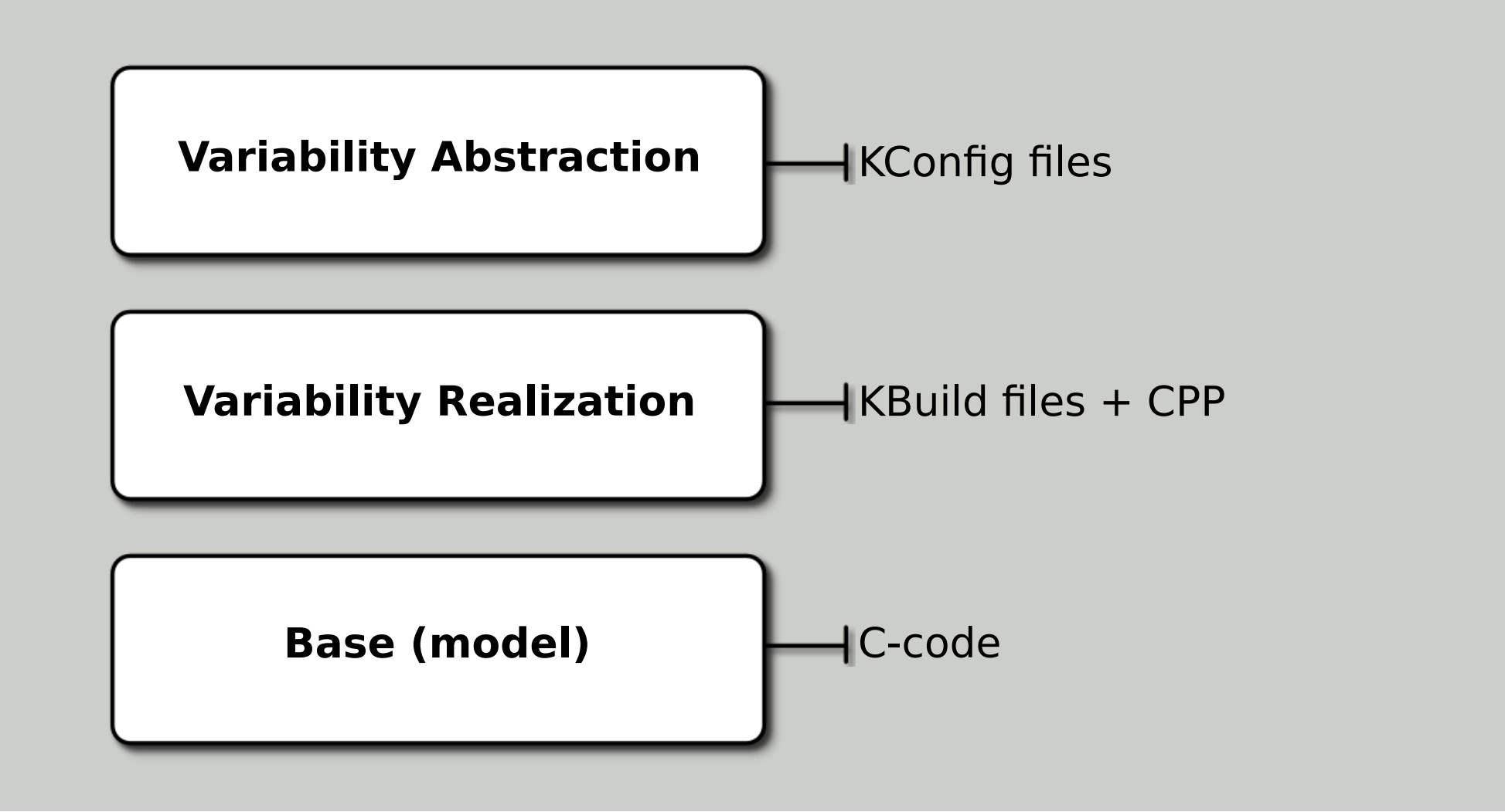
CVL Architecture for Dummies

The degree of coupling can be controlled by moving the mapping



CVL submitters. *Common Variability Language, OMG Revised Submission*. 2012

CVL Architecture for Linux Junkies



Grid View in Pure•Variants

pure·systems

Variant Management - CarLight/Configs/configspace.xml - Eclipse Platform

File Edit Navigate Search Project Run Variant Window Help

Functional_Features.xfm | *Configs

CarLight/Configs/configspace.xml

Model Elements	Level	Var1_00...	Var1_00...	Var1_00...	Var1_00...	Var1_	Var1_	Var1_	Var1_	Var1_	Var1_	Var1_00...	Var1_00...	Var1_00...	Var1_00...	Var1_00...	Var1_00...	Var1_00...	Var1_00...	Var1_00...	Var1_00...
Functional_Features																					
External Car ...		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Regions	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EMEA	1.1	✓	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓
EU	1.1.1	✓	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓
Austria	1.1.1.1	☐	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓
Denmark	1.1.1.2	☐	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓
UK	1.1.1.3	✓	✗	✗	✓	✗	☐	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓
North Ameri...	1.2	✗	✓	✓	✗	✓	✗	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗
Canada	1.2.1	✗	✓	✓	✗	✓	✗	☐	✓	✓	✗	✓	☐	✗	✓	✓	✓	✗	✗	✗	✗
Mexiko	1.2.2	✗	✓	✓	✗	✓	✗	✓	☐	✓	✗	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗
USA	1.2.3	✗	✓	✓	✗	✓	✗	✓	✓	✓	✗	✓	✓	✗	✓	✓	☐	✗	✗	✗	✗
Beam Confi...	2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Low Beam	2.1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Xenon	2.1.1	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Halogen	2.1.2	✗	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
High Beam	2.2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

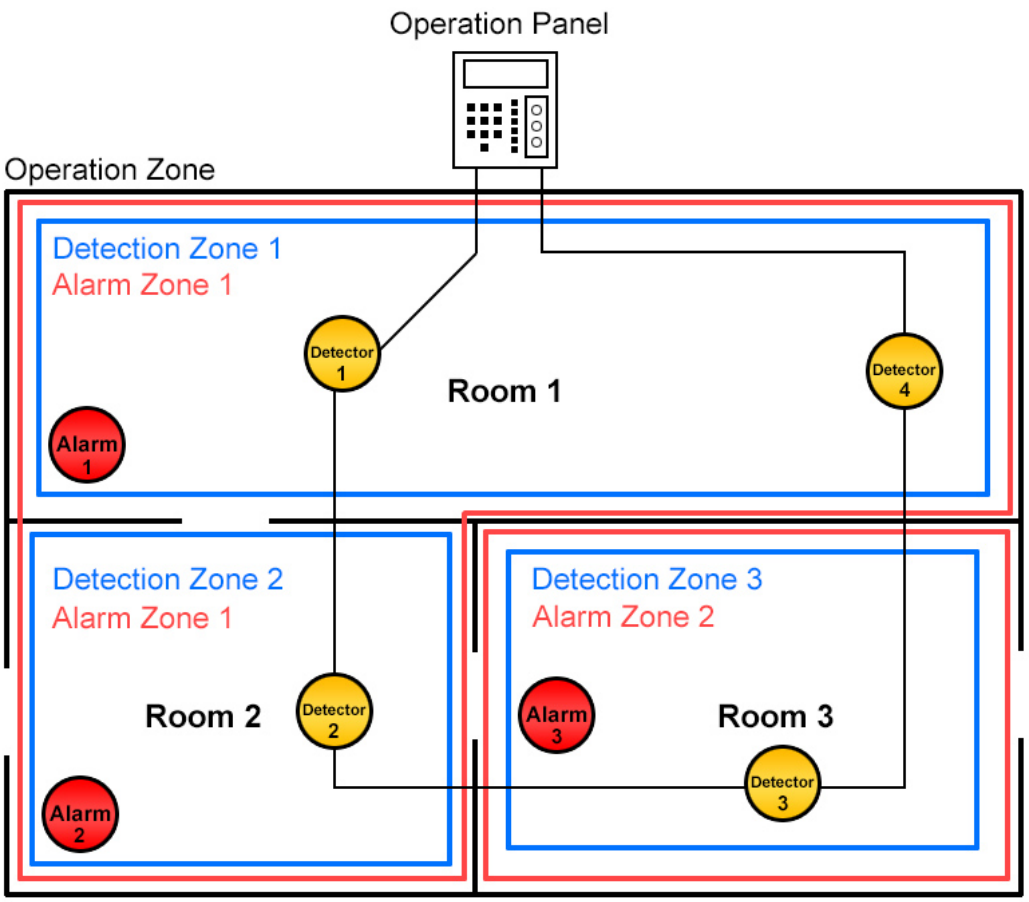
Matrix

Synchronize ClearQuest Defects

- Commercial tools support **multiple views** of the same model
- Some vendors: Pure Systems (DE), Big Lever (US), most PLM tools
- Clafer also has the grid view

Are all variability models trees?

A Fire Alarm System



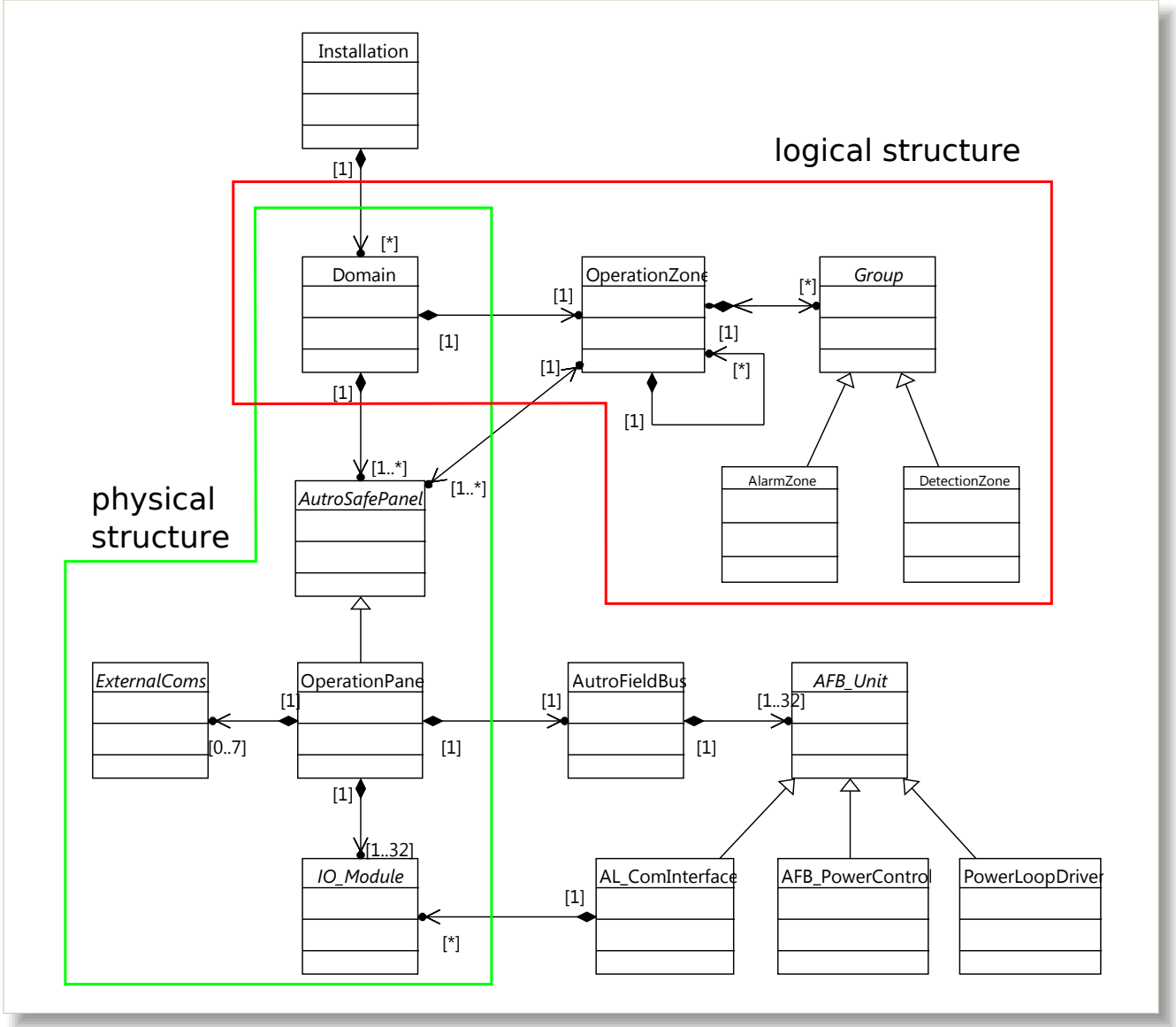
- detection zones
- alarm zones
- wiring
- three different structures
- Modeled as constrained class diagrams!
- Sometimes called **topological variability**

Berger. Stanciulescu. Øgård. Haugen. Larsen. Wąsowski. *To connect or not to connect: experiences from modeling topological variability.* SPLC 2014

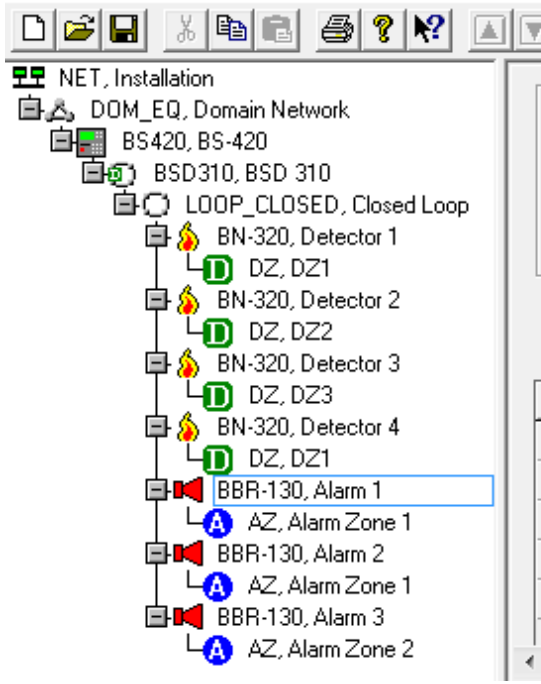
Fantechi. *Topologically configurable systems as product families.* SPLC 2013

Modeled using class diagrams in Papyrus

A model view showing two model hierarchies



A home grown **configurator** used to instantiate models

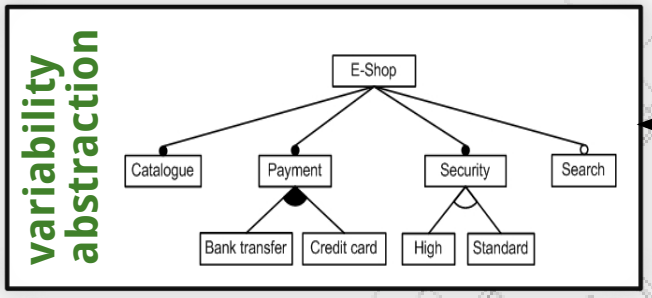


- SPL Method, **Architecture**
- Variability **Implementation Spectrum**
- Variability Abstraction: **Feature Modeling**
- Variability Modeling in **Practice**
- Variability **Realization**

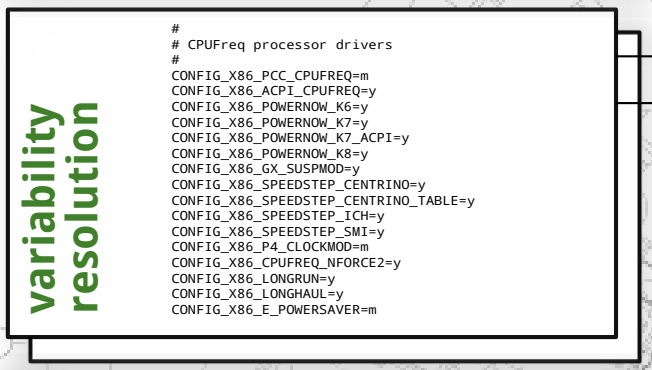
AGENDA

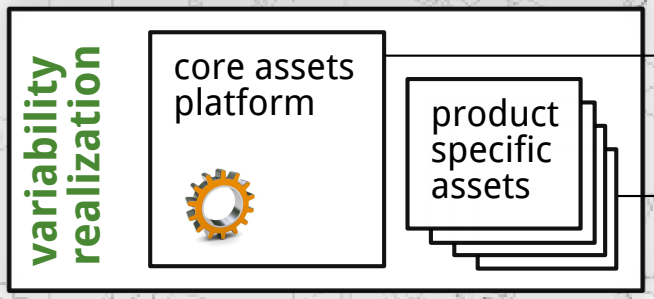
Connect Abstraction to Realization

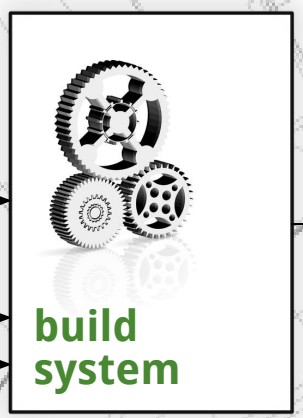
Most of the school is about it :)



configured against
conforms

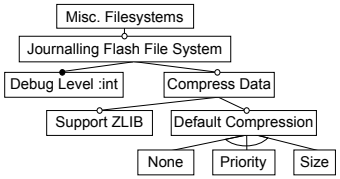




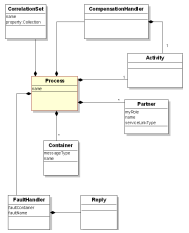


Feature Models vs DSLs

Feature Model



DSL Meta-Model



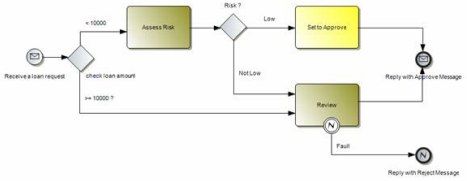
conforms to

conforms to

Configuration

```
#
# CPUFreq processor drivers
#
CONFIG_X86_PCC_CPUFREQ=m
CONFIG_X86_ACPI_CPUFREQ=y
CONFIG_X86_POWERNOW_K6=y
CONFIG_X86_POWERNOW_K7=y
CONFIG_X86_POWERNOW_K7_ACPI=y
CONFIG_X86_POWERNOW_K8=y
CONFIG_X86_GX_SUSPMOD=y
CONFIG_X86_SPEEDSTEP_CENTRINO=y
CONFIG_X86_SPEEDSTEP_CENTRINO_TABLE=y
CONFIG_X86_SPEEDSTEP_ICH=y
CONFIG_X86_SPEEDSTEP_SMI=y
CONFIG_X86_P4_CLOCKMOD=m
CONFIG_X86_CPUFREQ_NFORCE2=y
CONFIG_X86_LONGRUN=y
CONFIG_X86_LONGHAUL=y
CONFIG_X86_E_POWERSAVER=m
```

DSL Model

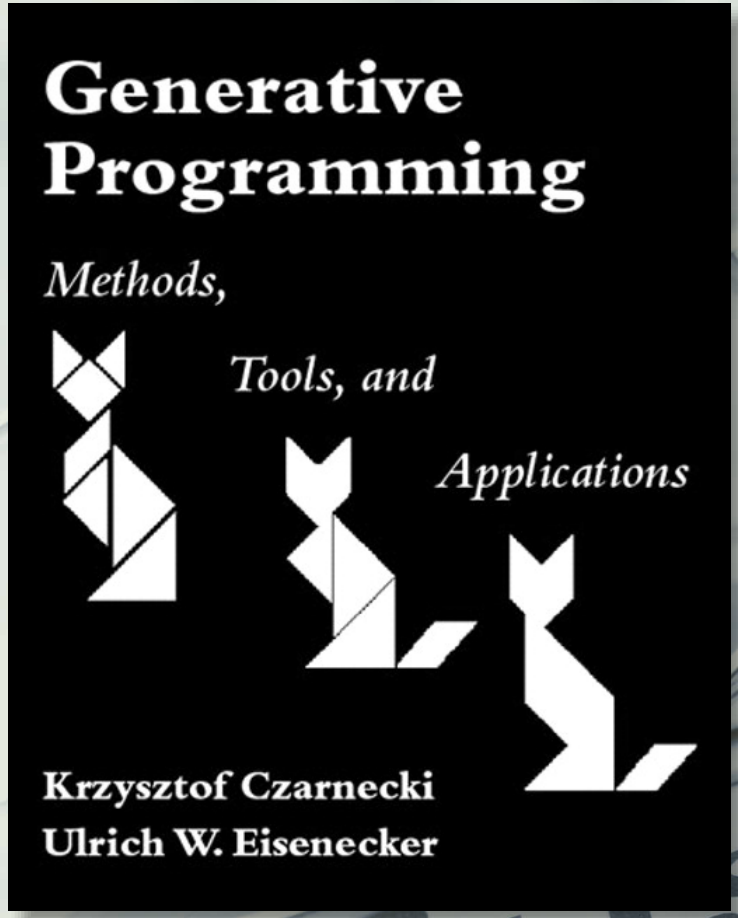


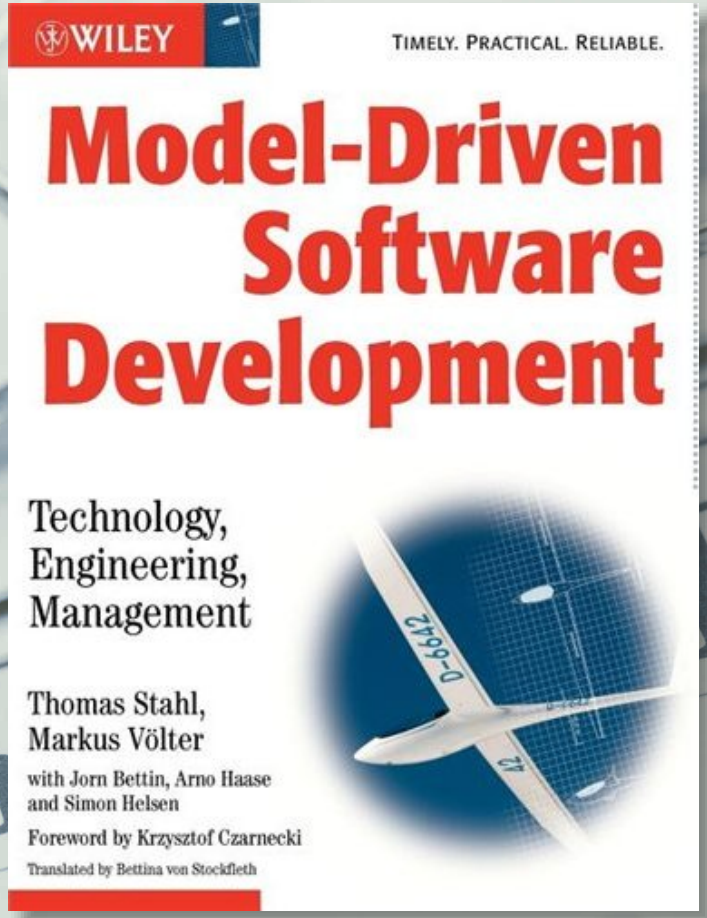
- Feature models are **ready and simple** (no design effort, deep insight)
- DSL requires **design effort**, but rewards with more **expressiveness**
- Effort also translates to **maintenance**
- FM effort is offset by **existing feature modeling tools**
- DSL development effort is offset by **language workbenches**

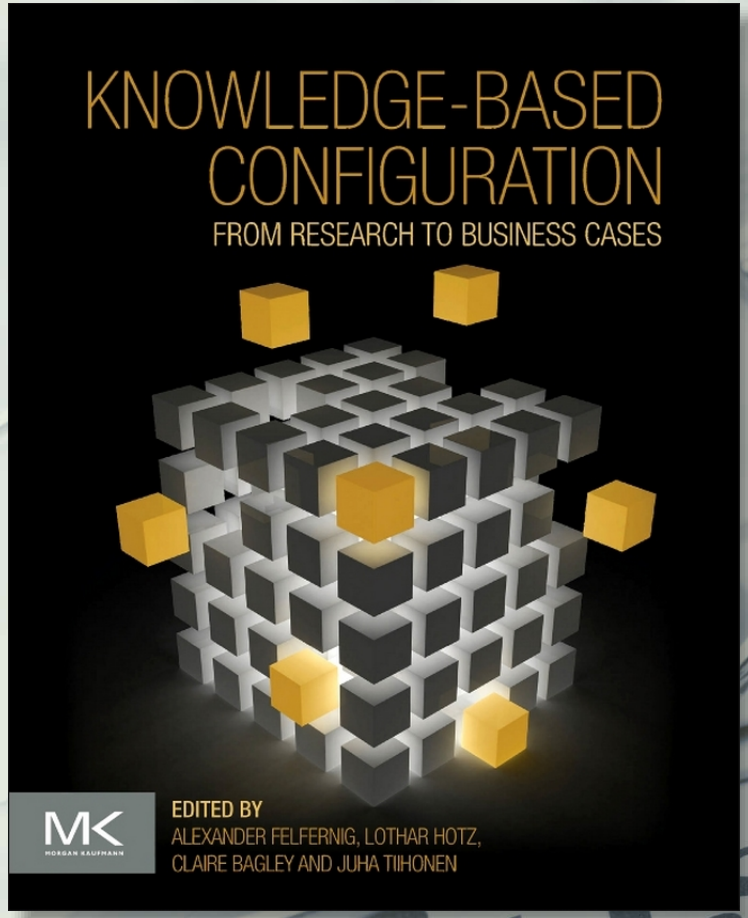
Advice on Realization

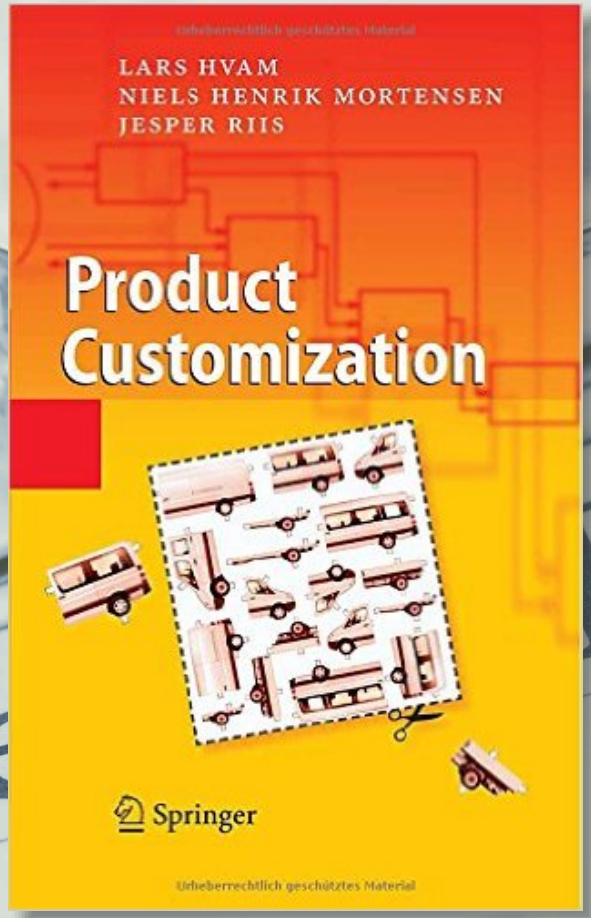
[Stahl and Völter]

- Choose **functional** domain **concepts** as features / DSL concepts
- Start a **small** domain model and **grow** it **iteratively**
- Keep the **build automatic** at all times
- Generate/synthesize **legible code/models**
- We follow these principles in the Clafer tutorial on railway stations


Generative Programming
Methods,
Tools, and
Applications
Krzysztof Czarnecki
Ulrich W. Eisenecker


WILEY
TIMELY. PRACTICAL. RELIABLE.
Model-Driven Software Development
Technology, Engineering, Management
Thomas Stahl, Markus Völter
with Jorn Bettin, Arno Haase and Simon Helsen
Foreword by Krzysztof Czarnecki
Translated by Bettina von Stockfleth


KNOWLEDGE-BASED CONFIGURATION
FROM RESEARCH TO BUSINESS CASES
MK
MORGAN KAUFMANN
EDITED BY
ALEXANDER FELFERNIG, LOTHAR HOTZ,
CLAIRE BAGLEY AND JUHA TIIHONEN


LARS HVAM
NIELS HENRIK MORTENSEN
JESPER RIIS
Product Customization
Springer

- SPL Method, **Architecture**
- Variability **Implementation Spectrum**
- Variability Abstraction: **Feature Modeling**
- Variability Modeling in **Practice**
- Variability **Realization**

AGENDA

FLANDERS MAKE

making the industry stronger and more competitive

Interested to join our team?
Find our open positions and apply at

https://jobs.flandersmake.be

FLANDERS MAKE

DRIVING INNOVATION IN MANUFACTURING

Bart Meyers

bart.meyers@flandersmake.be