



# Software Product Line Engineering

**modeling and managing variability  
of software intensive systems**

Dr. Mathieu Acher  
email: [macher@fundp.ac.be](mailto:macher@fundp.ac.be)

Prof. Patrick Heymans

University of Namur  
PReCISE Research Centre

# Material

- [http://www.fundp.ac.be/etudes/cours/page\\_view/INFOM435/](http://www.fundp.ac.be/etudes/cours/page_view/INFOM435/)
  - Folder: “Documents\_sur\_VariabilityAndSPL”
  - Slides, exercises, evaluation

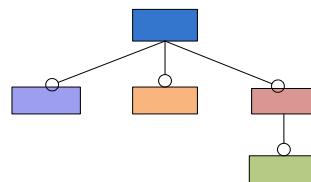
The screenshot shows a web-based course management interface. At the top, there's a header with links for 'Mon bureau', 'Liste de mes cours', 'Mon compte utilisateur', and 'Quitter'. Below the header, it displays the course information: 'Questions spéciales d'ingénierie du logiciel et de l'information INFOM435 - Vincent ENGLEBERT, Naji HABRA, Jean-Luc HAIAUT, Patrick HEYMANS'. The navigation bar includes 'Mode d'affichage', 'Etudiant', and 'Gestionnaire de cours'. The main content area is titled 'Documents et liens' and contains a table with the following data:

Nom	Taille	Date	Modifier	Supprimer	Déplacer	Visibilité
CoursAgile			X	X	X	X
CoursQualité			X	X	X	X
documentation_MoCQA			X	X	X	X
documents_sur_VariabilityAndSPL			X	X	X	X
Software product line engineering (variability modeling and management)			X	X	X	X
Syllabus						

- Email: [macher@fundp.ac.be](mailto:macher@fundp.ac.be)

# Previously

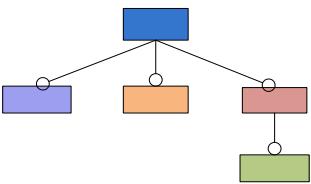
- Feature Models (FMs)
- Practice
  - Existing car configurators
  - TVL (FM language)
  - FAMILIAR (FM management)



Feature Model

# Previously

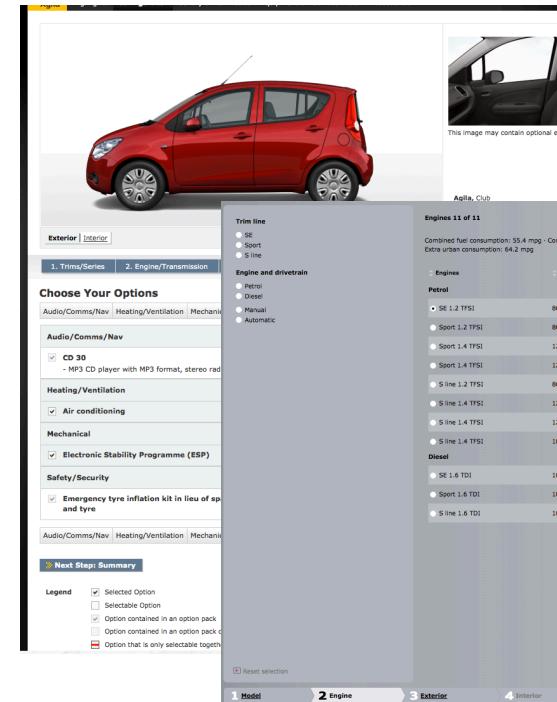
Language (TVL, FAMILIAR)



Feature Model



Automated reasoning



The screenshot shows a car configuration interface for an Audi Club model. At the top, there's a red car thumbnail and a side-view camera thumbnail. Below the car is a large double-headed horizontal arrow pointing left and right, indicating a relationship between the Feature Model and the configuration interface. The main area contains a configuration menu with the following sections:

- Choose Your Options**:
  - Trim line: Sport, S line
  - Engine and drivetrain:
    - Petrol
      - SE 1.2 TFSI
      - Sport 1.2 TFSI
      - Sport 1.4 TFSI
      - S line 1.2 TFSI
      - S line 1.4 TFSI
      - S line 1.4 TFSI
      - S line 1.6 TDI
      - Sport 1.6 TDI
      - S line 1.6 TDI
    - Diesel
      - SE 1.6 TDI
      - Sport 1.6 TDI
      - S line 1.6 TDI
  - Audio/Comms/Nav
  - Heating/Ventilation
  - Mechanical
  - Safety/Security

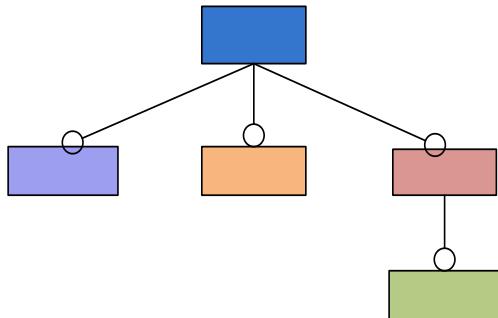
At the bottom, there's a "Next Step: Summary" button and a legend for option status indicators:

  - Selected Option (checkmark)
  - Selectable Option (empty box)
  - Option contained in an option pack (checkmark)
  - Option contained in an option pack (empty box)
  - Option that is only selectable together (red box)

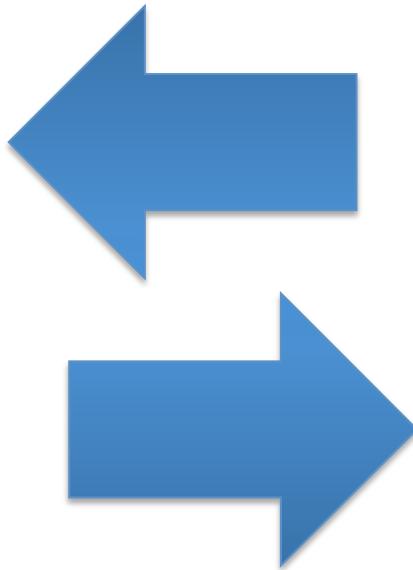
Navigation buttons at the bottom include: 1 Model, 2 Engine, 3 Exterior, 4 Interior.

# Running project

- Re-engineer a car configurator
  - we are making some progress, right?



Variability Model  
(Feature Model)



The screenshot shows a car configurator interface for an Audi Agila Club. On the left, there's a large image of a red three-door hatchback. To its right is a smaller image of the interior. Below the car are two sections: 'Trim line' and 'Engines 11 of 11'. The 'Trim line' section lists 'SE', 'Sport', and 'S line' options. The 'Engines' section lists various engine variants with their power, torque, and price. At the bottom, there's a legend for option selection status and a 'Next Step: Summary' button.

Trim line

- SE
- Sport
- S line

Engines 11 of 11

Engine	Power (PS)	Gearbox	Drive train	RRP (GBP)
SE 1.2 TFSI	86	5 speed	Front-wheel drive	13,335.00
<input checked="" type="radio"/> Sport 1.2 TFSI	86	5 speed	Front-wheel drive	15,175.00
Sport 1.4 TFSI	122	6 speed	Front-wheel drive	15,585.00
Sport 1.4 TFSI	122	6 speed	Front-wheel drive	17,035.00
S line 1.2 TFSI	86	5 speed	Front-wheel drive	16,720.00
S line 1.4 TFSI	122	6 speed	Front-wheel drive	17,130.00
S line 1.4 TFSI	122	6 speed	Front-wheel drive	18,580.00
S line 1.4 TFSI	185	5 speed	Front-wheel drive	20,510.00
Diesel				
<input checked="" type="radio"/> SE 1.6 TDI	105	5 speed	Front-wheel drive	14,395.00
Sport 1.6 TDI	105	5 speed	Front-wheel drive	16,235.00
S line 1.6 TDI	105	5 speed	Front-wheel drive	17,780.00

Legend

- Selected Option
- Selectable Option
- Option contained in an option pack
- Option contained in an option pack
- Option that is only selectable together with other options

Reset selection

1 Model    2 Engine    3 Exterior    4 Interior    5 Equipment    6 Your Audi

# Today

- Back to our modeling experience
  - Feature model management (FAMILIAR)
    - Support to better understand and play with your specification
    - Existing techniques can be considered in your work (merging)
    - Building views (slicing)
- Project
  - Re-engineering configurators
    - « reverse engineering » + « engineering »
  - Now: From feature models to configurators
    - Model-based approach

# Open issues

- Quality of the feature model?
  - Completeness
  - Correctness

## Support for validation

de laboratoire. Par conséquent, notre modèle TVL n'est pas complet dans le sens où tous les configuteurs n'ont pas été modélisées. Nous avons par contre bien compris la base de la modélisation TVL.

Quant à son exactitude, nous ne pouvons être sûr de nous. D'une part, nous sommes loin d'être des experts de la modélisation TVL, une meilleure implémentation est sans doute possible, d'autre part, nous savons juste, grâce au parser disponible, que notre code est syntaxiquement correct, mais nous ne savons comment nous pouvons le tester ou l'exécuter. A première vue et selon nous, notre code nous semble exact.

An advantage of the tool is the possibility to validate the created model with the little Java program. You can then see whether your features model is correct (by example, if the model is syntactically correct) or not, or how many possible instances you can have and so on.

### 2.3 Validation

Right now, we have no tools for validating the built diagram. The syntax errors are reported with a little of meaningful information on what is wrong with the file. Also, typing errors are very hard to spot when there is just one different letter.

Visualizing the built feature diagram could be helpful if the diagram isn't too big. For bigger models, listing the possible configurations when some of the features are restrained could be helpful to debug the TVL.

A photograph of an old, rusted green pickup truck. The truck is heavily damaged, with its front end crushed and the driver's side door missing. It is situated in a field of tall, dry grass and weeds, with a dense thicket of bushes in the background.

Unused flexibility

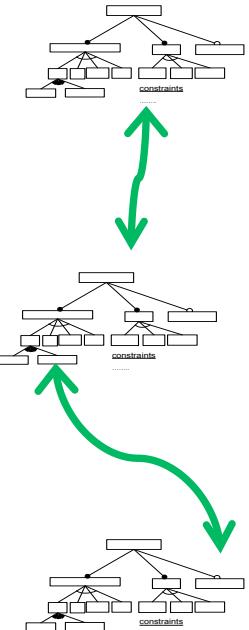
A police car is engulfed in large flames, with fire visible from the front and rear. The car is white with blue and red stripes on the side. The word "POLICE" is written on the front grille and "To Serve & Protect" is written on the side door. In the background, there are buildings with signs for "GUESS", "DRUM SHOP", and "KIRK'S".

DRUM SHOP

JAS  
STORE

# Illegal Variant

# FAMILIAR language and environment

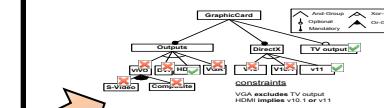


Interoperability

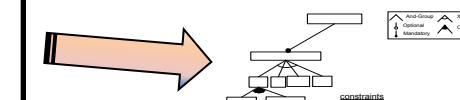
```
// foo.fml
fm1 = FM ("foo1.tvl")
fm2 = FM ("foo2.m")
fm3 = merge intersection { fm1 fm2 }
c3 = counting fm3
renameFeature fm3.TV as "OutputTV"
fm5 = aggregate { fm3 FM ("foo4.xml") }
assert (isValid fm5)
fm6 = slice fm5 including fm5.TV.*
export fm6
```

FAMILIAR

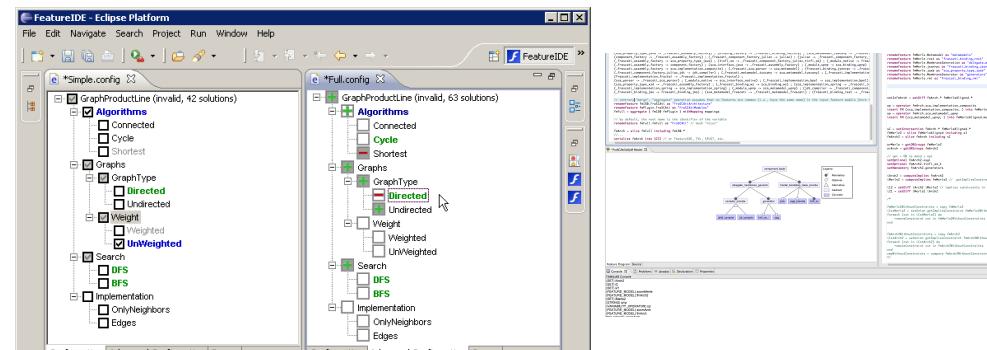
Language facilities



True/False  
8759  
"OutputTV", "TV"



Environment



# Support to better understand and play with your specification

- Error detection
  - Void feature model
  - “Anomalies”
    - Dead features
    - False Optionals
    - Redundant constraints
  - Edit your FMs and replay the operations
- Interactive session
  - Configure your FM
  - Assert some properties

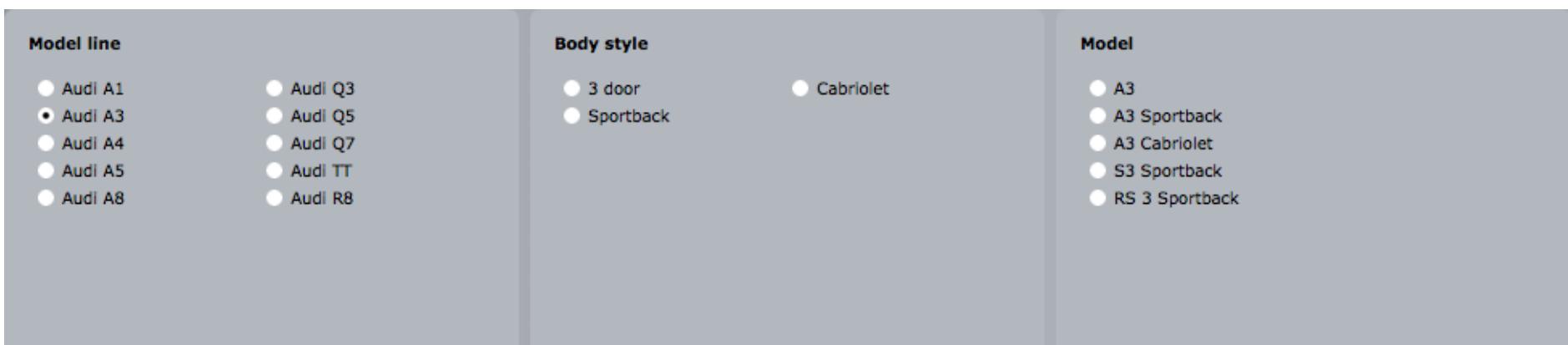
# Problem: multiple „car models“

The image shows a screenshot of an Audi website. At the top, there are three rows of Audi cars: Row 1 has A1 (red), A3 (black), and A4 (silver); Row 2 has A5 (dark grey), A8 (black), Q3 (black), and Q5 (silver); Row 3 has Q7 (black), TT (silver), and R8 (black). The Audi logo and slogan "Vorsprung durch Technik" are in the top right. Below the cars, there's a search bar with "Enter Audi Code" and a "Next" button. The main content area is titled "Model line" and lists two columns of car models:

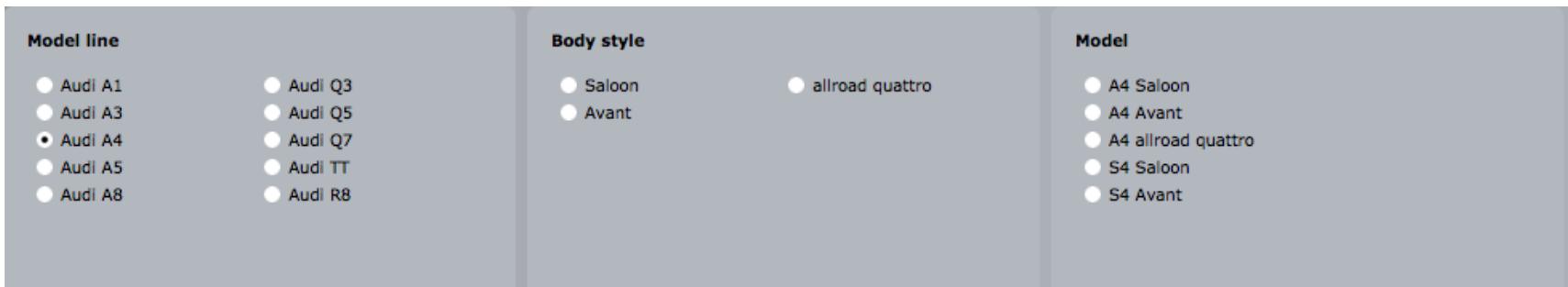
Model line	Body style	Model
Audi A1		Audi Q3
Audi A3		Audi Q5
Audi A4		Audi Q7
Audi A5		Audi TT
Audi A8		Audi R8

At the bottom, a navigation bar shows steps 1 through 6: 1 Model, 2 Engine, 3 Exterior, 4 Interior, 5 Equipment, and 6 Your Audi. There are also "Sitemap" and "Terms of Use" links.

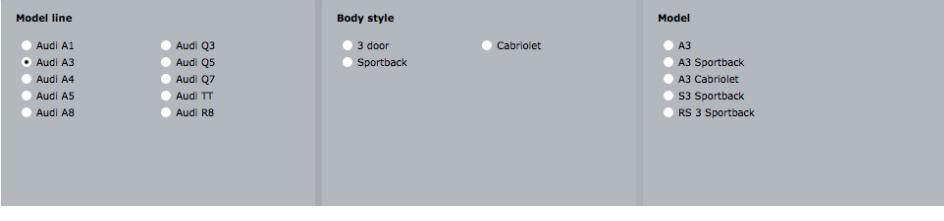
# Problem: multiple „car models“



# Problem: multiple „car models“



# Problem: multiple „car models“



## Two modeling approaches

**#1 – top-down: specify constraints (e.g., excludes) for each model line all along selection/deselection**

**#2 – bottom-up: elaborate an FM for each model line and merge them**

# #1 top-down

## Model line

- Audi A1
- Audi A3
- Audi A4
- Audi A5
- Audi A8
- Audi Q3
- Audi Q5
- Audi Q7
- Audi TT
- Audi R8

## Body style

- Saloon
- Avant
- 3 door
- Sportback
- Coupé
- Cabriolet
- Roadster
- Spyder
- allroad quattro
- SUV

## Model

- A1
- A1 Sportback
- A3
- A3 Sportback
- A3 Cabriolet
- S3 Sportback
- RS 3 Sportback
- A4 Saloon
- A4 Avant
- A4 allroad quattro
- S4 Saloon
- S4 Avant
- A5 Coupé
- A5 Sportback
- A5 Cabriolet
- S5 Coupé
- S5 Sportback
- S5 Cabriolet
- RS 5 Coupé
- A8
- A8L
- A8 W12
- Q3
- Q5
- Q7
- TT Coupé
- TT Roadster
- TTS Coupé
- TTS Roadster
- TT RS Coupé
- TT RS Roadster
- R8 Coupe
- R8 Spyder

```
ModelLine {
group oneof
AudiA1 {
    AudiA1 -> (A1 || A1Sportback);
    AudiA1 -> (Door3 || Sportback);
},
AudiA3 {
    AudiA3 -> (A3 || A3Sportback || A3Cabriolet || S3 || S3Sportback || RS3Sportback);
    AudiA3 -> (Door3 || Sportback || Cabriolet);
},
AudiA4 {
    AudiA4 -> (A4Saloon || A4Avant || A4AllroadQuattro || S4Saloon || S4Avant);
    AudiA4 -> (Saloon || Avant || AllroadQuattro);
},
AudiA5 {
    AudiA5 -> (A5Coupe || A5Sportback || A5Cabriolet || S5Coupe || S5Sportback || S5Cabriolet || RSSCoupe);
    AudiA5 -> (Sportback || Coupe || Cabriolet);
},
AudiA6 {
    AudiA6 -> (A6Saloon || A6Avant);
    AudiA6 -> (Saloon || Avant);
},
```

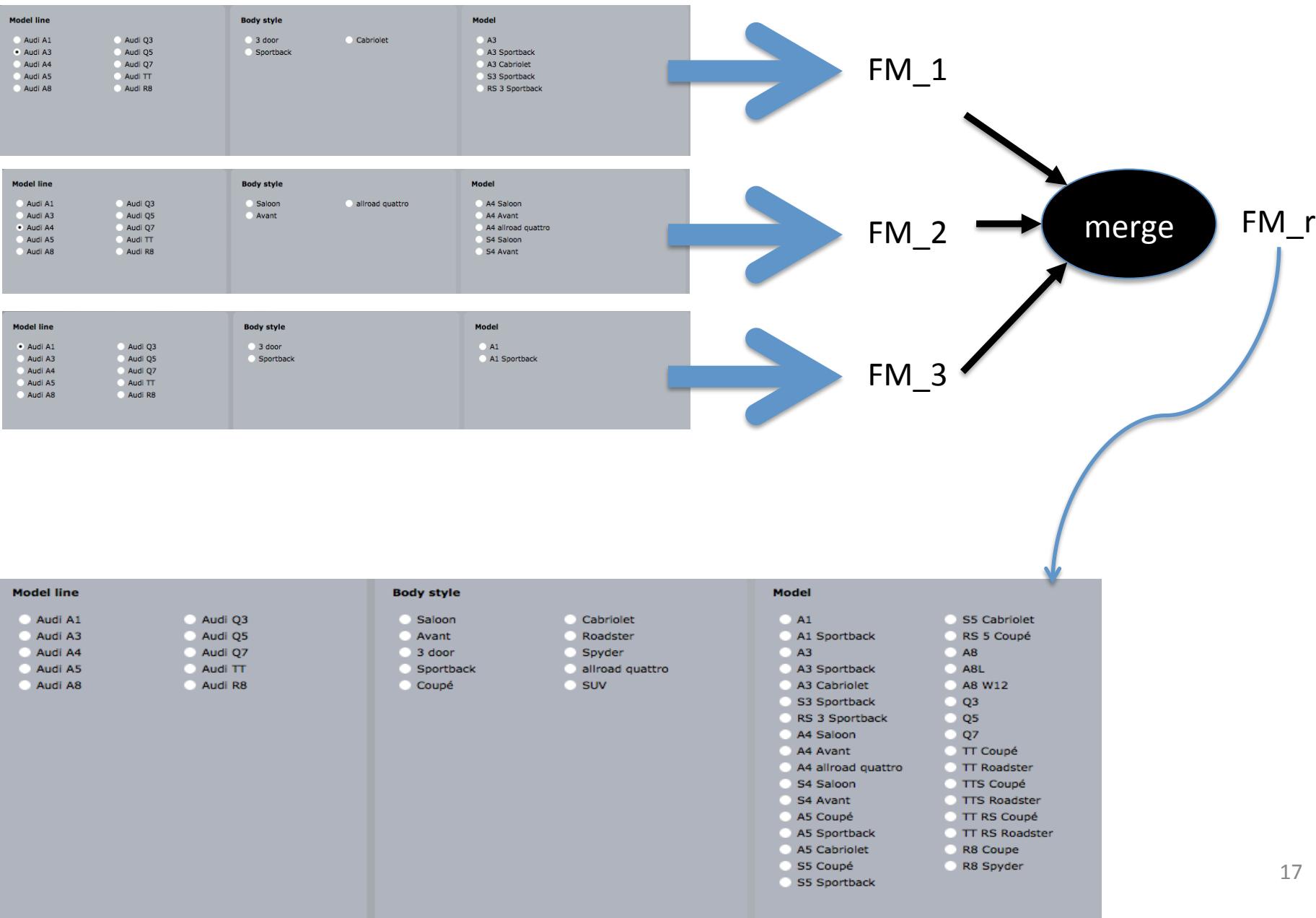


## BodyStyle {

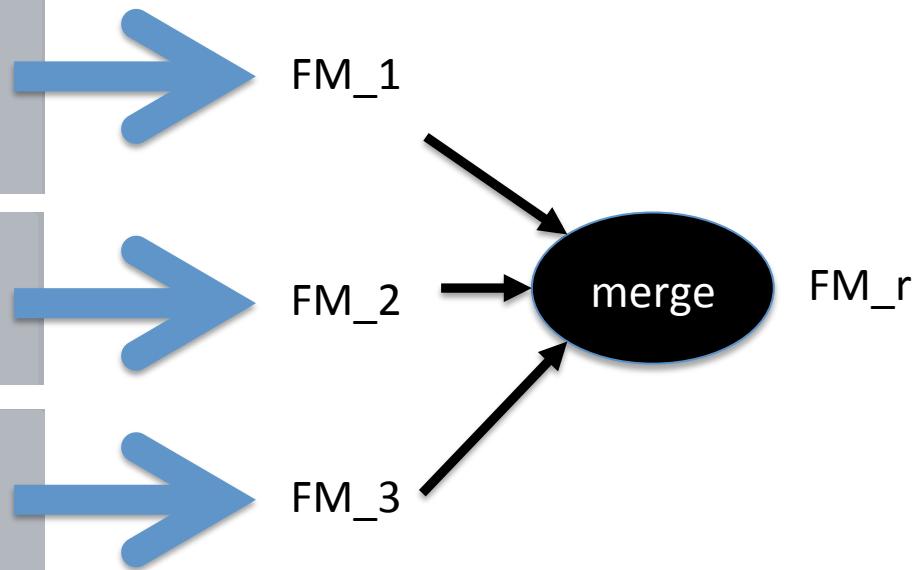
```
group oneof {
    Saloon
    {
        Saloon -> (A4Saloon || S4Saloon || A6Saloon || A8 || A8L || A8W12);
    },
    Avant
    {
        Avant -> (A4Avant || S4Avant || A6Avant);
    },
    Door3
    {
        Door3 -> (A1 || A3 || S3);
    },
    Sportback
    {
        Sportback -> (A1Sportback || A3Sportback || S3Sportback || RS3Sportback || A5Sportback || S5Sportback || A7Sportback);
    },
    Coupe
    {
        Coupe -> (A5Coupe || S5Coupe || RSSCoupe || TTCCoupe || TTSCoupe || TTRSCoupe || R8Coupe);
    },
}
```



# #1 bottom-up



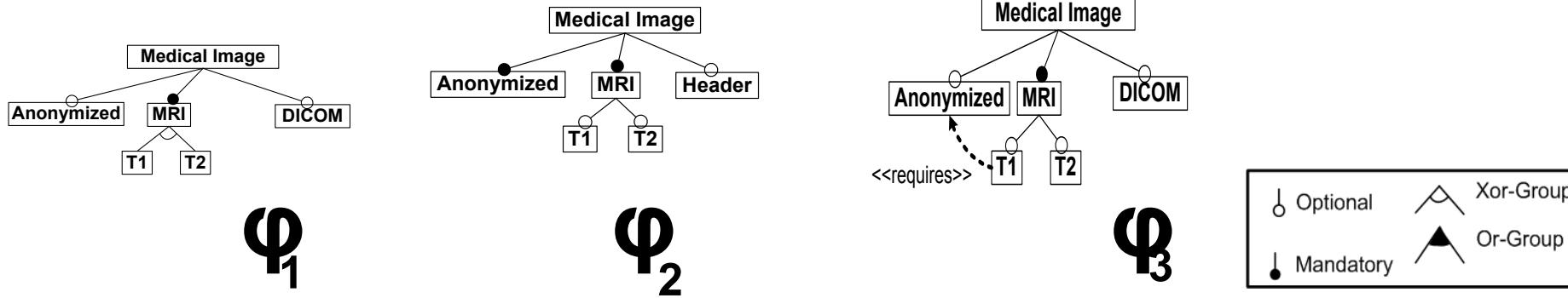
# #1 bottom-up (FAMILIAR)



```
a fml> fmAudiS = merge sunion { fm1 fm2 fm3 }
a fmAudiS: (FEATURE_MODEL) Audi: ModelLine BodyStyle ;
F ModelLine: (A1|A4|A3) ;
L BodyStyle: (Saloon|Door3)? (Cabriolet|AllroadQuattro)? (Sportback|Avant)? ;
f (A4 -> !Sportback);
f (A1 -> !Avant);
M (A1 -> !Saloon);
E (A3 -> !Saloon);
f (A4 -> !Cabriolet);
N (A1 -> !AllroadQuattro);
E (A1 -> !Cabriolet);
f (A4 -> !Door3);
f (A3 -> !Avant);
M (A3 -> !AllroadQuattro);
E
```

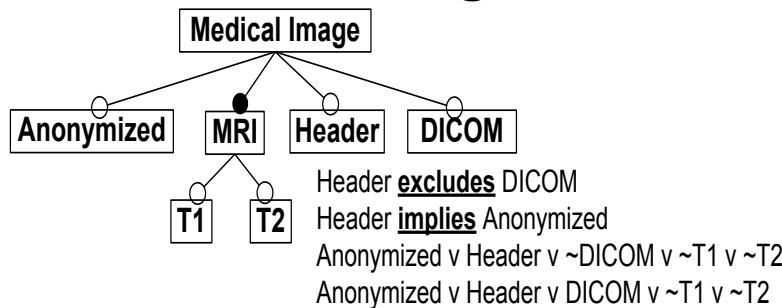
9.9.4 (beta)

# Merging operation: an example



$\Phi_{123}$   
merged propositional formula

How to synthesise a  
feature model that  
represents the union of  
input sets of  
merged hierarchy  
configurations?



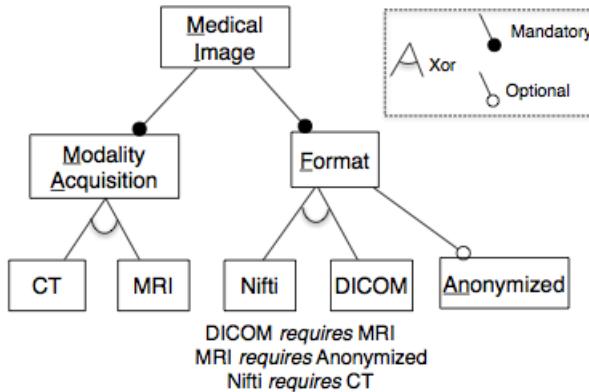
Set mandatory features  
Detect Xor and Or-groups  
Compute “implies/excludes”  
constraints

# Merging operation: algorithm (ideas)

```

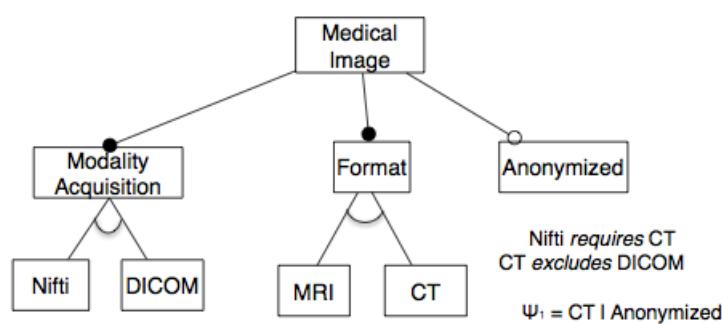
 $s_0 = \{$ 
     $\{MI, MA, F, CT, Nifti\},$ 
     $\{MI, MA, F, CT, Nifti, AN\},$ 
     $\{MI, MA, F, DICOM, MRI, AN\}$ 
 $\}$ 

```

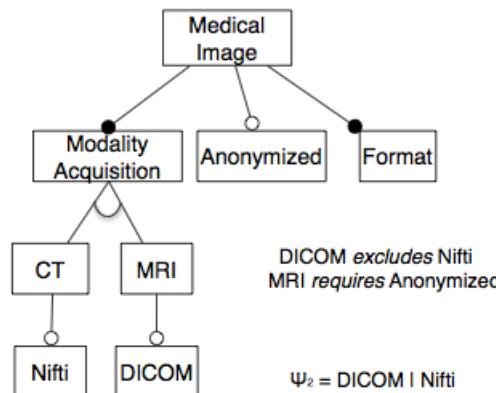


(a) set of configurations

(b)  $fm_0$



(c)  $fm_1$



(d)  $fm_2$

Fig. 2: For a given set of configurations, three possible yet different FMs ( $s_0 = \llbracket fm_0 \rrbracket = \llbracket fm_1 \rrbracket = \llbracket fm_2 \rrbracket$ )

# Merging operation: algorithm (ideas)

K-SYNTH ( $\phi$  : formula, K : specification)

- 1 simplify  $\phi$  by removing dead features
  - 2 compute the binary implication graph (BIG)
  - 3 compute atomic sets (AS)
  - 4 compute binary exclusion graph (BEG)
- preprocessing steps*
- 5 setting feature hierarchy and mandatory features
    - $G = (\mathcal{F}, E, r)$  is a tree,  $\mathcal{F}$ : set of non dead features
    - $E_{\text{MAND}}$
  - 6 compute all possible Mutex-, Xor- and Or-groups  
setting feature groups
    - $G_{\text{MUTEX}}$ ,  $G_{\text{OR}}$ ,  $G_{\text{XOR}}$
  - 7 setting constraints
    - EQ, RE, EX
    - compute  $\Psi$

In line 3, we use *BIG* to identify *atomic sets*, i.e., set of features that are mutually implied and always appear together in configurations. There are identified as the strongly connected components in *BIG* [14, 10]. We denote *AS* the set of atomic sets of the formula  $\phi$ . In line 4, we compute the binary exclusion graph, noted *BEG* of the formula

# Merge vs Aggregate

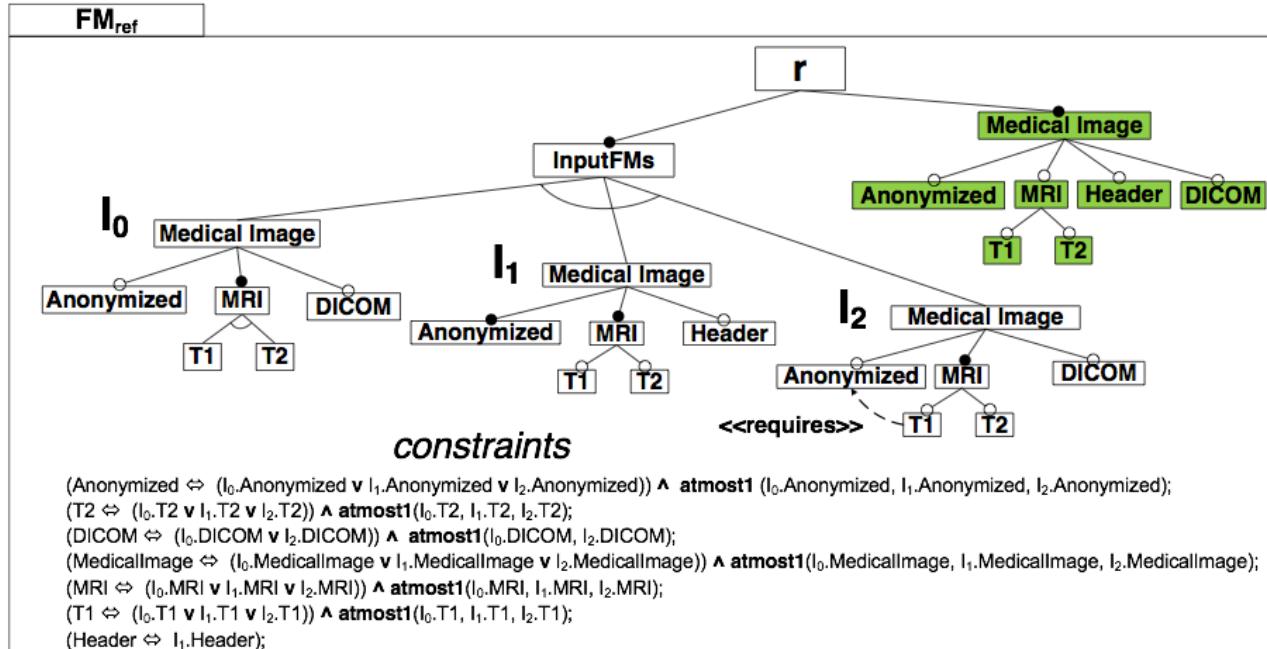
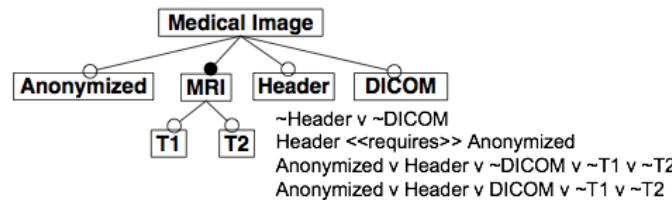


Figure 7.10: Merge of three feature models,  $I_0$ ,  $I_1$  and  $I_2$  using the slicing operator



# Building “views” of a feature model

```

not Audi car
group all{
    ModelLine;
    Model;
    ModelType;
    Model;
    Exterior;
    Interior;
    Equipment;
}
3

ModelLine {
    group all{
        AudiA1 {
            AudiA1 {
                AudiA1 -> Q1 || AllSportback || A3Cabriolet || S3 || S3Sportback || RS3Sportback;
                AudiA1 -> Q3 || Sportback || Cabriolet;
            }
            AudiA2 {
                AudiA2 -> Q2 || AllSportback || A3Cabriolet || S3 || S3Sportback || RS3Sportback;
                AudiA2 -> Q2r || Sportback || Cabriolet;
            }
            AudiA3 {
                AudiA3 -> (A3Saloon || A3Avant || AllroadQuattro || S4Saloon || S4Avant);
                AudiA3 -> (Q3e | Coupé || Avant || AllroadQuattro);
            }
            AudiA4 {
                AudiA4 -> (Q5Coupé || AllSportback || A4Cabriolet || S5Coupé || S5Sportback || S5Cabriolet || RS5Coupé);
                AudiA4 -> (Sportback || Coupé || Cabriolet);
            }
            AudiA5 {
                AudiA5 -> (Q5Cabriolet || A5Avant);
                AudiA5 -> (Coupé || Avant);
            }
            AudiA6 {
                AudiA6 -> (Q7 || All || A6M2);
                AudiA6 -> (Saloon);
            }
            AudiA7 {
                AudiA7 -> (Q7);
                AudiA7 -> (RS7);
            }
            AudiA8 {
                AudiA8 -> (Q8 || TFSI);
                AudiA8 -> (Coupé || R8);
            }
        }
        AudiS1 {
            AudiS1 -> (A1);
            AudiS1 -> (A1);
        }
        AudiS2 {
            AudiS2 -> (A1);
            AudiS2 -> (A1);
        }
        AudiS3 {
            AudiS3 -> (A3);
            AudiS3 -> (A3);
        }
        AudiS4 {
            AudiS4 -> (A4);
            AudiS4 -> (A4);
        }
        AudiS5 {
            AudiS5 -> (A5);
            AudiS5 -> (A5);
        }
        AudiS6 {
            AudiS6 -> (A6);
            AudiS6 -> (A6);
        }
        AudiS7 {
            AudiS7 -> (A7);
            AudiS7 -> (A7);
        }
        AudiS8 {
            AudiS8 -> (A8);
            AudiS8 -> (A8);
        }
        AudiRS {
            AudiRS -> (A1);
            AudiRS -> (A1);
        }
        AudiRS3 {
            AudiRS3 -> (A3);
            AudiRS3 -> (A3);
        }
        AudiRS4 {
            AudiRS4 -> (A4);
            AudiRS4 -> (A4);
        }
        AudiRS5 {
            AudiRS5 -> (A5);
            AudiRS5 -> (A5);
        }
        AudiRS6 {
            AudiRS6 -> (A6);
            AudiRS6 -> (A6);
        }
        AudiRS7 {
            AudiRS7 -> (A7);
            AudiRS7 -> (A7);
        }
        AudiRS8 {
            AudiRS8 -> (A8);
            AudiRS8 -> (A8);
        }
        AudiTT {
            AudiTT -> (TFSI);
            AudiTT -> (Roadster);
        }
        AudiTTS {
            AudiTTS -> (TFSI);
            AudiTTS -> (Roadster);
        }
        AudiTTRS {
            AudiTTRS -> (TFSI);
            AudiTTRS -> (Roadster);
        }
        AudiTTRS {
            AudiTTRS -> (TFSI);
            AudiTTRS -> (Roadster);
        }
        AudiR8 {
            AudiR8 -> (Coupé || Spyder);
            AudiR8 -> (Coupé || Spyder);
        }
    }
}

bodyStyle {
    group all{
        Coupé {
            Coupé -> (A3Saloon || S4Saloon || A6Saloon || A8 || AB || ABM2);
        };
        Avant {
            Avant -> (A3Sportback || S4Sportback || A6Sportback || AB || ABM2);
        };
        Cabriolet {
            Cabriolet -> (A3Cabriolet || AS4Cabriolet || S5Cabriolet);
        };
        Roadster {
            Roadster -> (TFSI);
            Roadster -> (TFSI);
            Roadster -> (TFSI);
            Roadster -> (TFSI);
        };
        Spyder {
            Spyder -> (TFSI);
            Spyder -> (TFSI);
        };
    }
}

```

The image shows a section of the Audi website featuring a grid of Audi vehicles. The models displayed are A1, A3, A4, A8, Q3, Q5, Q7, TT, and R8. Below the cars, a list of Audi models is provided, each with a brief description of its variants:

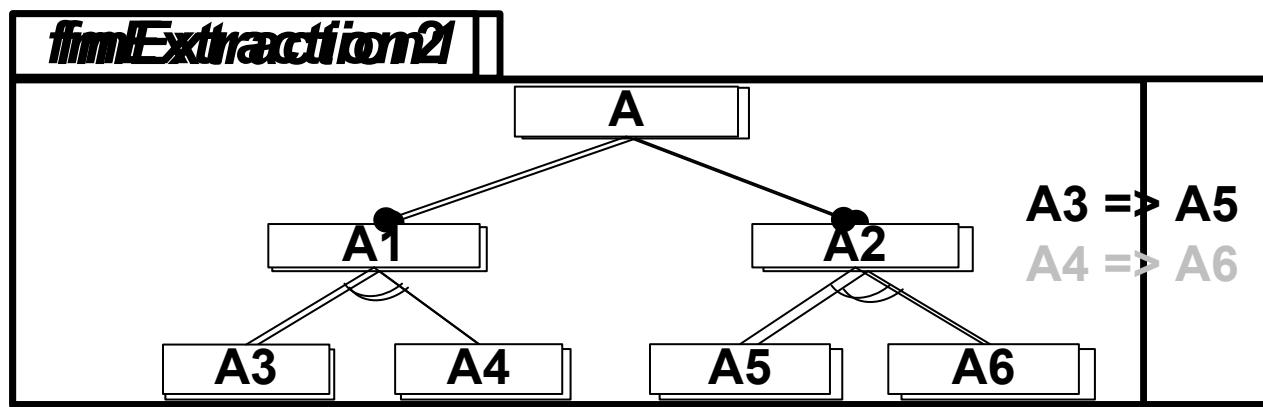
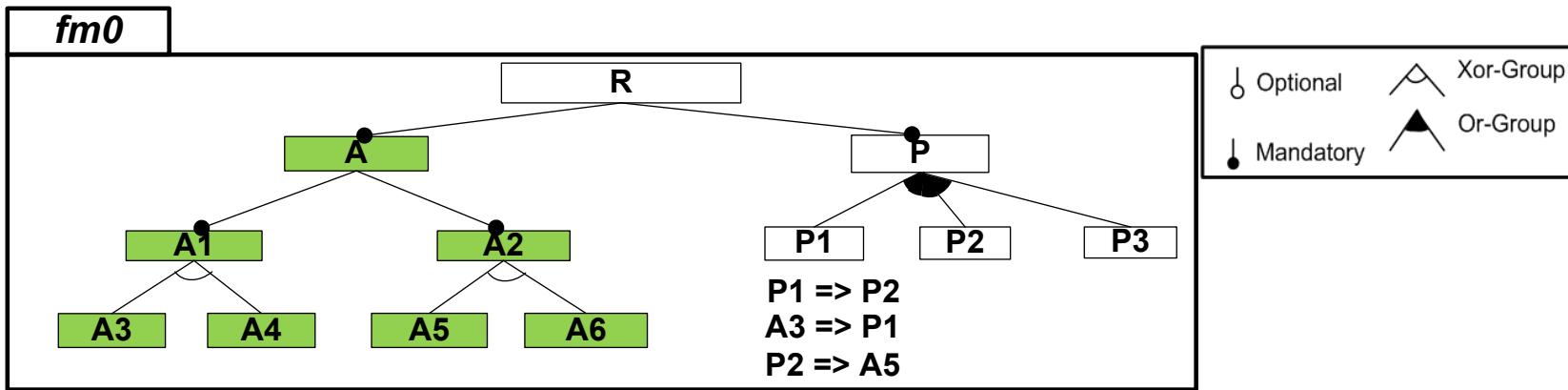
- A5 Cabriolet**: RS 5 Coupé
- A6**: A8
- A8L**: A8 W12
- A8 W12**: Q3
- Q5**: Q7
- Q7**: TT Coupé
- TT Roadster**: TT Roadster
- TTS Coupé**: TT RS Coupé
- TT RS Coupé**: TT RS Roadster
- TT RS Roadster**: R8 Coupé
- R8 Coupé**: R8 Spyder
- R8 Spyder**: R8 Spyder

At the bottom of the page, there is a navigation bar with six tabs labeled 1 Model, 2 Engine, 3 Exterior, 4 Interior, 5 Equipment, and 6 Your Audi. Below the navigation bar, there is a link to "Sitemap" and "Terms of Use".

# Building “views” of a feature model

- Problem: given a feature model, how to decompose it into smaller feature models?
- Semantics?
  - What’s the hierarchy
  - What’s the set of configurations?

# A first try

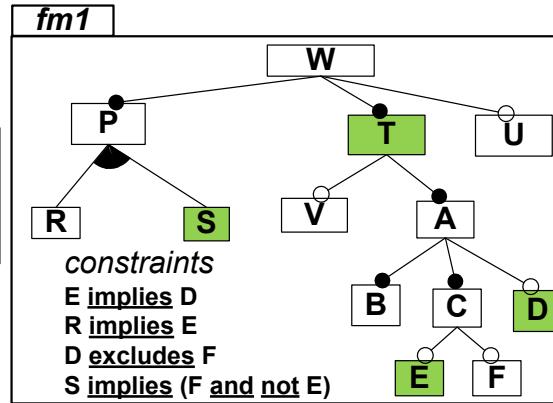
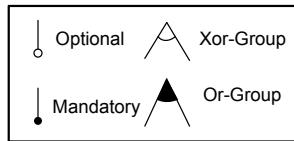


**Problem:** You can select A3 without A5

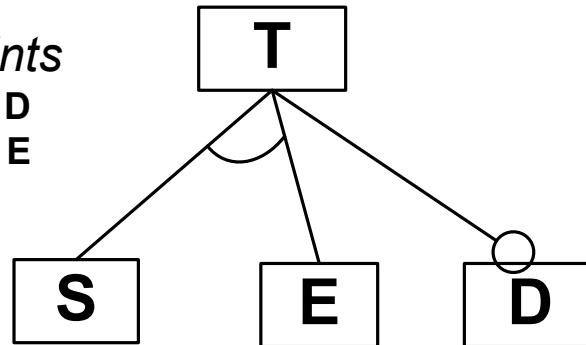
# Hierarchy and Configuration matter!

# Slicing Operator

**slicing criterion** arbitrary set of features, relevant for a feature model user



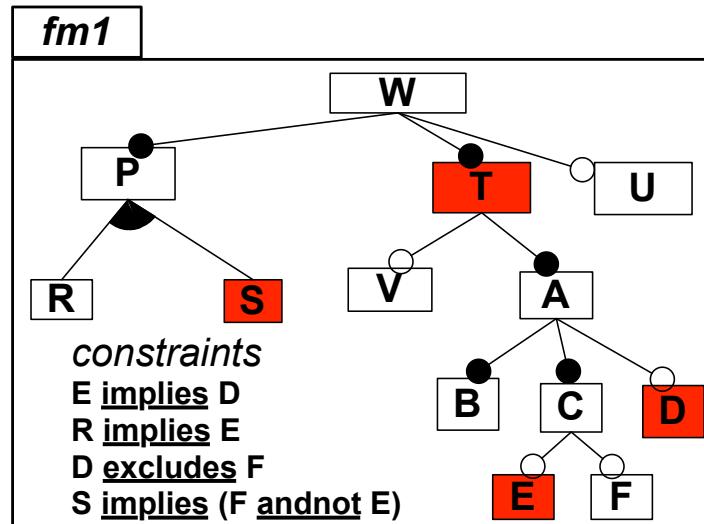
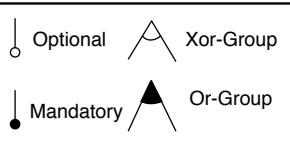
*constraints*  
**E implies D**  
**D implies E**



**slice** a new feature model, representing a projected set of configurations

# Slicing operator: going into details

## projected set of configurations



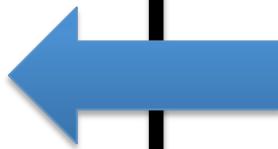
```
fm1 != {  
  {A,B,C,D,E,R,T,U,V},  
  {A,B,C,E,P,S,T,U,V},  
  {A,B,C,D,E,R,T,U,V},  
  {A,B,C,E,P,S,T,U,V},  
  {A,B,C,E,P,S,T,U,V},  
  {A,B,C,E,P,S,T,U,V},  
  {A,B,C,D,E,R,T,U,V},  
}
```

```
fm1p = {  
  {D,E,T},  
  fm1p = {  
    {S,T},  
    {B,E,T},  
    {S,T},  
    {S,T},  
    {S,T},  
    {D,E,T}  
  }  
}
```

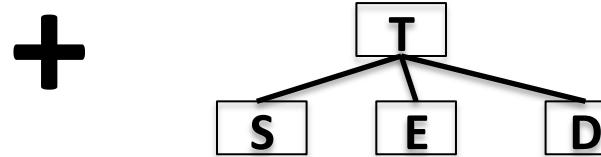
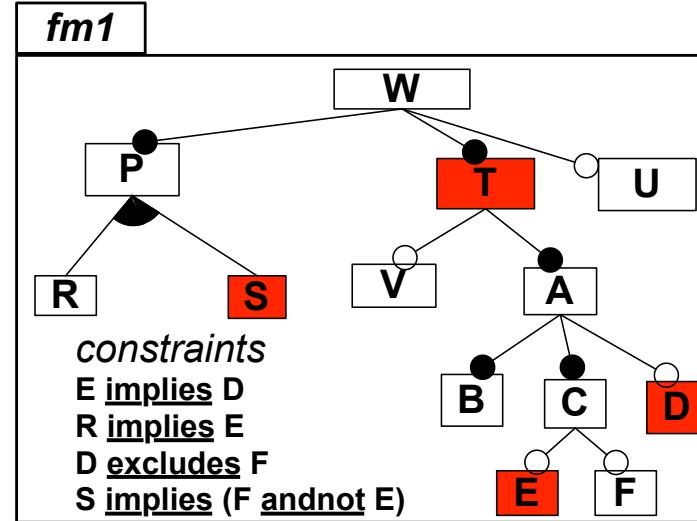
# Slicing operator: going into details

## synthesizing the corresponding feature model

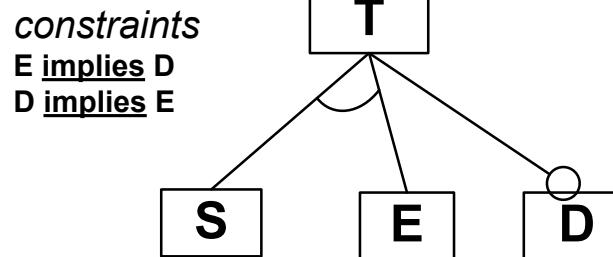
$\varphi_1$   
↓  
 $\varphi_{s1}$



existential quantification  
of features  
not included  
in the slicing criterion



fm1p = {  
  {D,E,T},  
  {S,T}  
}



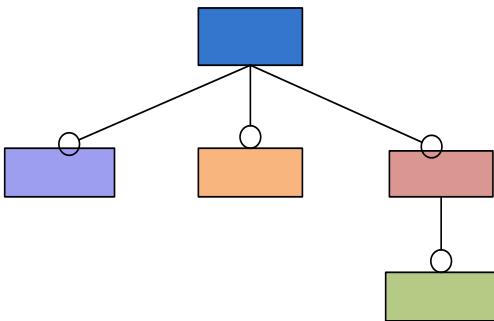
# Back to our modeling experience

- Not easy (error-prone, time-consuming)
  - Support to better understand and play with your specification
  - Automated techniques can be considered (merging)
- Feature model management to the rescue
  - Also: building views (slicing)
- Open issues for reverse engineering feature models
  - « variability patterns »
  - automated support

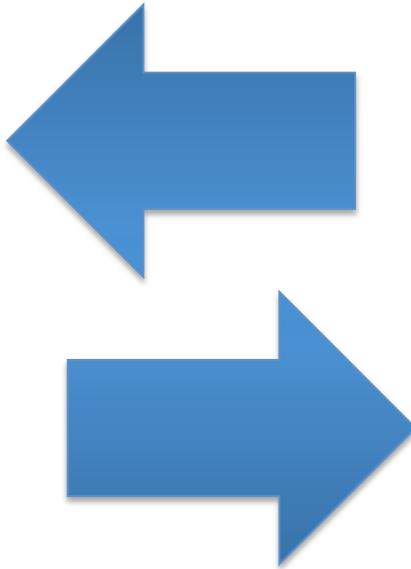
# Today

- Back to our modeling experience
  - Feature model management (FAMILIAR)
    - Support to better understand and play with your specification
    - Existing techniques can be considered in your work (merging)
    - Building views (slicing)
- Project
  - Re-engineering configurators
    - « reverse engineering » + « engineering »
  - Now: From feature models to configurators
    - Model-based approach

# Running project



**Variability Model  
(Feature Model)**



The screenshot shows a configuration interface for an Audi car. At the top, there's a red car model and a small interior view. Below the car, the text "Aquila, CUB" is visible. The main area contains a list of options under categories like "Trim line", "Engine and drivetrain", and "Choose Your Options". The "Engines 11 of 11" section lists various engine models with their power, torque, and price. The bottom of the screen shows a navigation bar with tabs for "Model", "Engine", "Exterior", "Interior", "Equipment", and "Your Audi".

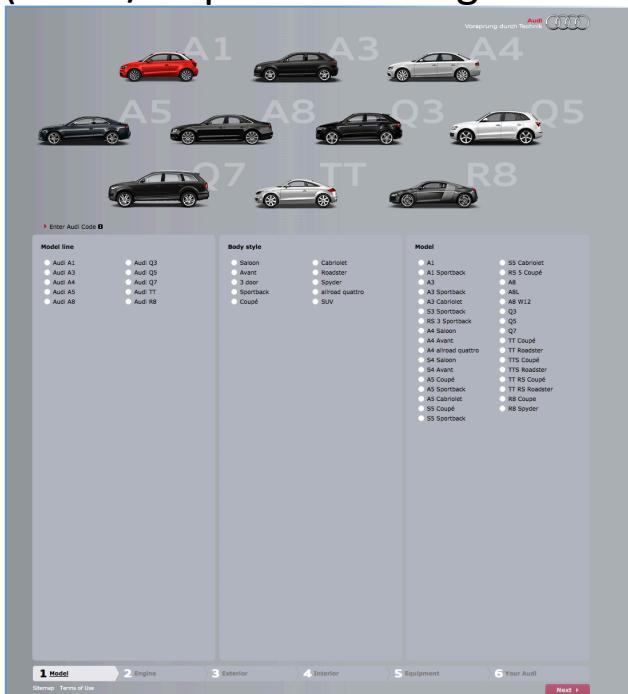
Engine	Power (PS)	Gearbox	Drivetrain	RRP (GBP)
SE 1.2 TFSI	86	5 speed	Front-wheel drive	13,335.00
Sport 1.2 TFSI	86	5 speed	Front-wheel drive	15,175.00
Sport 1.4 TFSI	122	6 speed	Front-wheel drive	15,585.00
Sport 1.4 TFSI	122	S tronic	Front-wheel drive	17,035.00
S line 1.2 TFSI	86	5 speed	Front-wheel drive	16,720.00
S line 1.2 TFSI	86	6 speed	Front-wheel drive	17,130.00
S line 1.4 TFSI	122	6 speed	Front-wheel drive	18,580.00
S line 1.4 TFSI	122	S tronic	Front-wheel drive	20,510.00
Diesel				
SE 1.6 TDI	105	5 speed	Front-wheel drive	14,395.00
Sport 1.6 TDI	105	5 speed	Front-wheel drive	16,235.00
S line 1.6 TDI	105	5 speed	Front-wheel drive	17,780.00

# Running project

- 70% of the final mark
- Expected output
  - Configurator artefacts (source code, metamodels/models, demo)
  - Presentation
    - “defense”
- Good properties
  - Solution that works
  - Reusable artefacts
  - Customizable solution
  - Close to existing configurator (“re-engineering”)
    - Or even better: “better”

# Running project

- 1 configurator by group
  - the one you previously chose
- Expected
  - (mandatory) all configuration options in one place
  - (optional) “views”
    - Different containers / tabs / pages
    - Wizards (linear/sequential configuration process)



# From feature models to configurators

- Configurator?
  - Graphical user interface (GUI)
  - Assist users
    - Select/deselect options
    - Propagation of choices
- Feature model?
  - Representation of the configuration “space”
  - Not a visually attractive solution!
- Filling the gap

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

<input type="radio"/> S	from € 19,495.00
<input checked="" type="radio"/> SC	from € 21,195.00
<input type="radio"/> SRI	from € 22,495.00
<input type="radio"/> SE	from € 23,495.00
<input type="radio"/> Elite	from € 25,495.00

Standard Equipment Filter: Heating/Ventilation

Heating/Ventilation

- Air conditioning with particle filter, manual controls

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

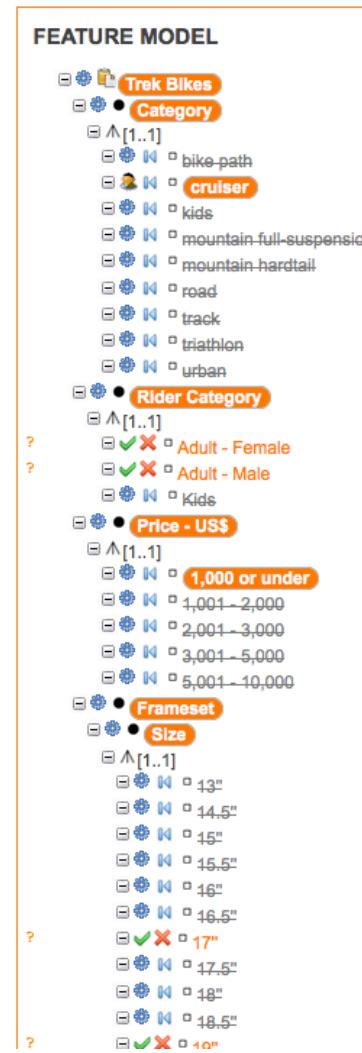
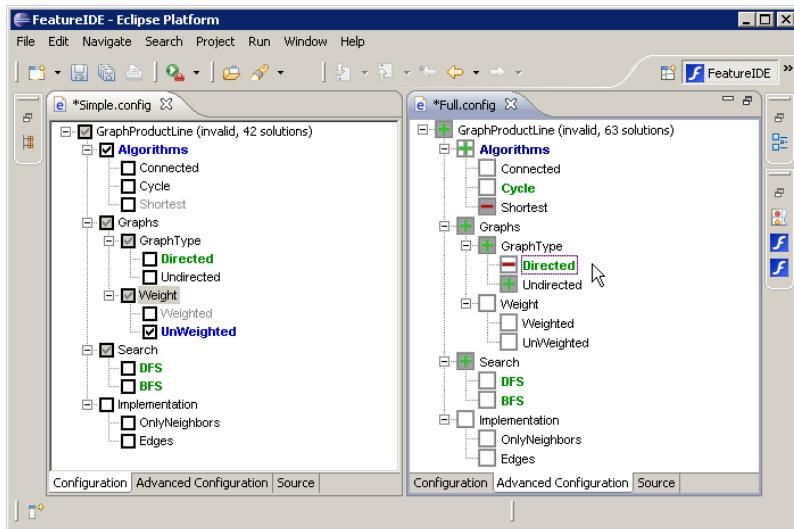
Interior	Comfort/Convenience	Option Packs	<b>Safety/Security</b>	Seating	Audio/Comms/Nav	
Heating/Ventilation	Mechanical	A-Z				

**Safety/Security**

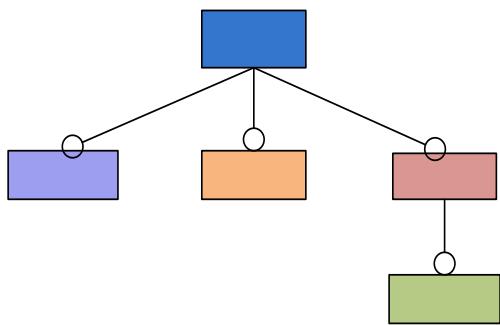
<input checked="" type="checkbox"/> Emergency tire inflation kit	Standard
<input type="checkbox"/> Active front seat head restraints	€ 103.00
- also adds seat belt unfastened warning light for front passenger's seat	
<input type="checkbox"/> Front and rear parking distance sensors	€ 407.00
<input checked="" type="checkbox"/> Tyre Pressure Monitoring System	€ 155.00
<input type="checkbox"/> Remote control ultrasonic security alarm system	€ 357.00

# From feature models to configurators

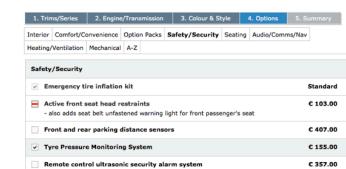
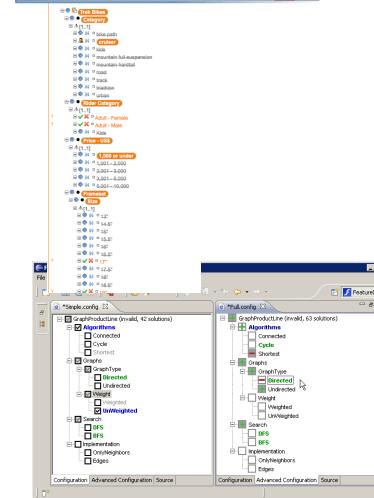
- Something like this



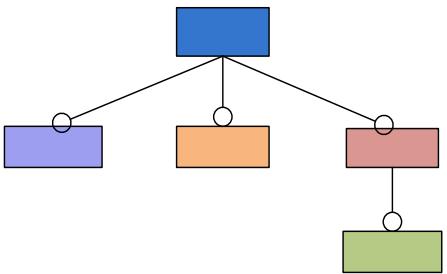
# Generic, flexible solution



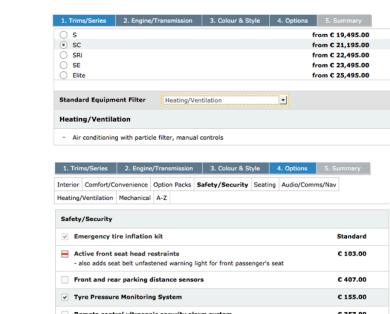
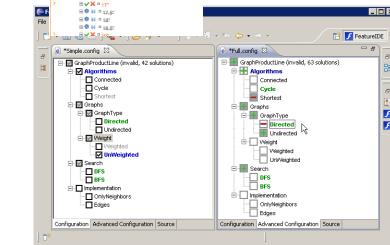
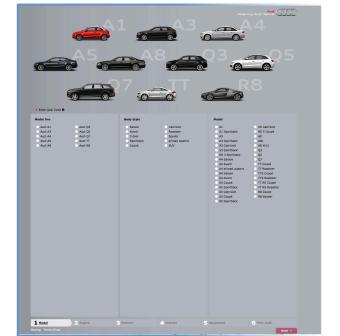
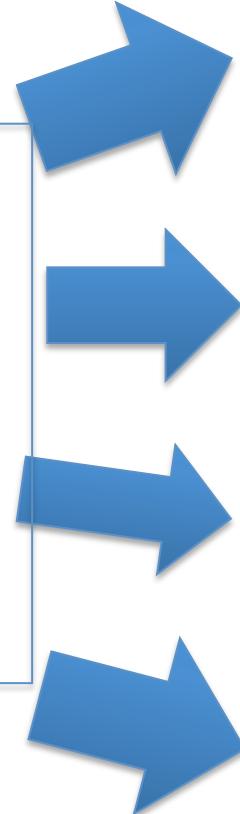
Feature Model



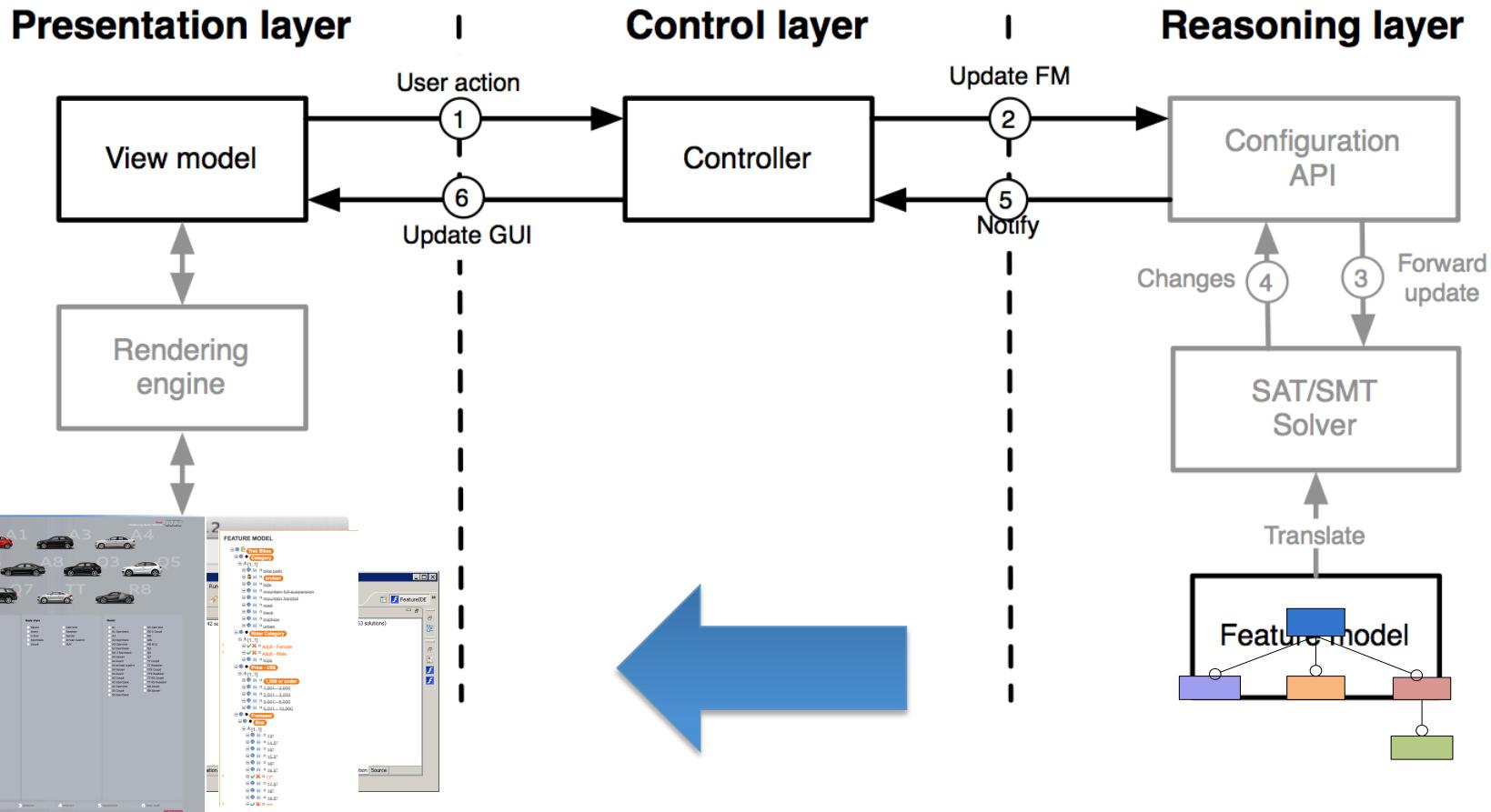
# Model-based approach



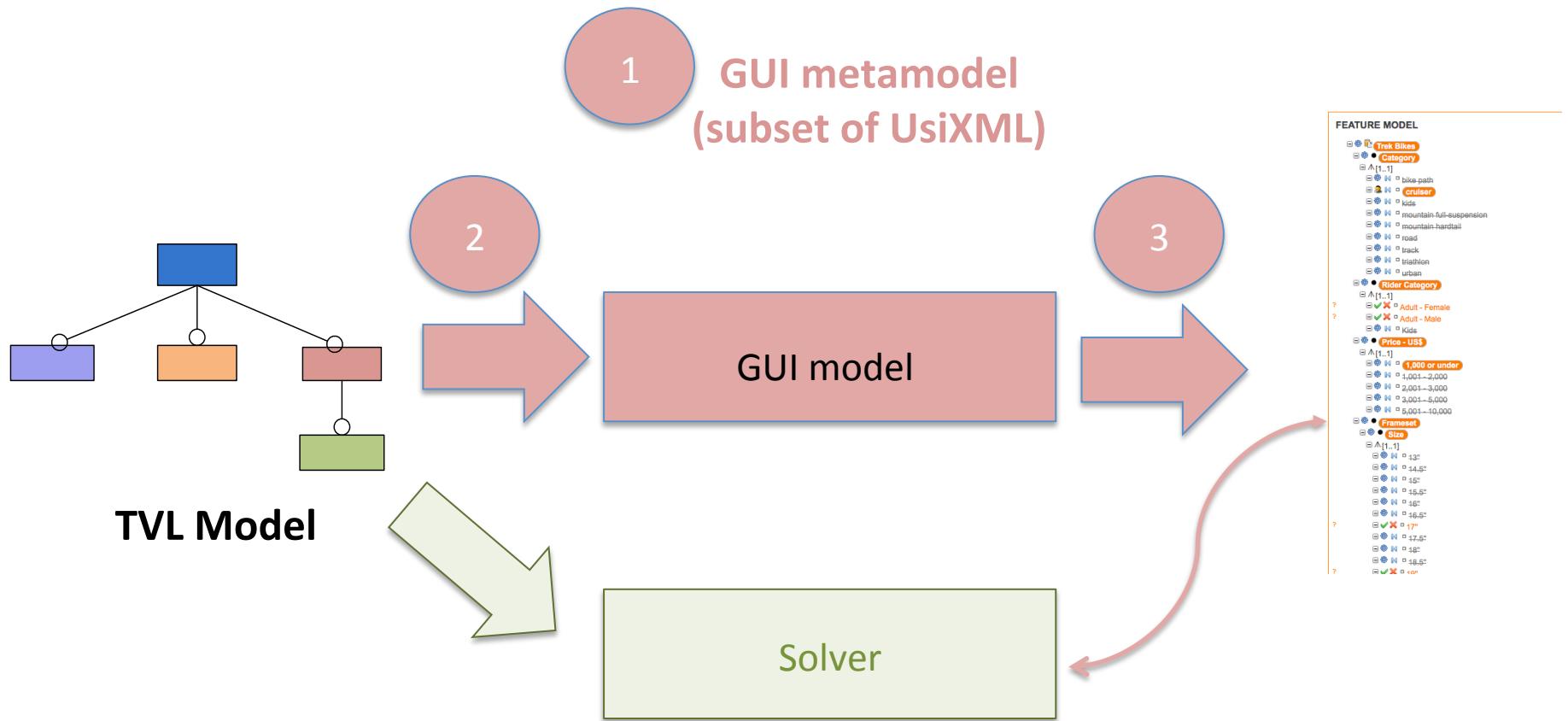
- #1 Configuration GUI model
- #2 Solver instance
- #3 Rendering and styling directives
- #4 ...



# MVC Architecture



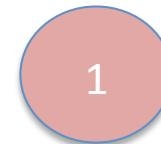
# Model-based approach



**Hierarchy:** rooted tree

**Variability:**

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



## GUI metamodel (subset of UsiXML)

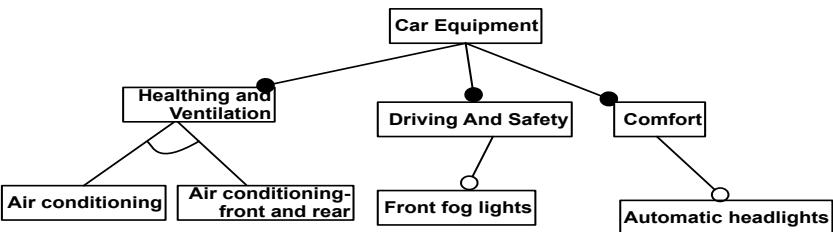


**RENAULT VANS**

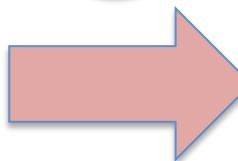
CARS | VANS | ELECTRIC VEHICLES | RENAULT BUSINESS | USED CARS | OWNERS

Renault UK > Renault Vans > New Kangoo Van Range > Kangoo Van > E

### TVL-2-GUIModel Transformation



2



### NEW KANGOO VAN RANGE

01 Preferences      02 Version

< Previous

Next >

#### OPTIONS

##### > AUDIO EQUIPMENT

4x20W radio CD MP3 Bluetooth with aux point and integrated display £125.00

##### > COMFORT

Height adjustable drivers seat £50.00

Climate control £800.00

Central storage console & armrest between seats £50.00

##### > DRIVING

Automatic headlights & wipers £100.00

Glazed Tailgate and rear wiper £150.00

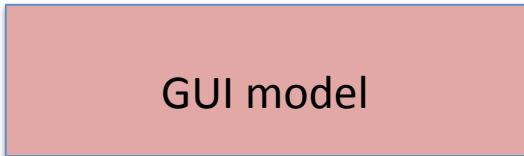
Engine Speed Limiter (56mph/90kmh) £50.00

Engine Speed Limiter (62mph/100kmh) £50.00

# Concrete GUI (one technological space)

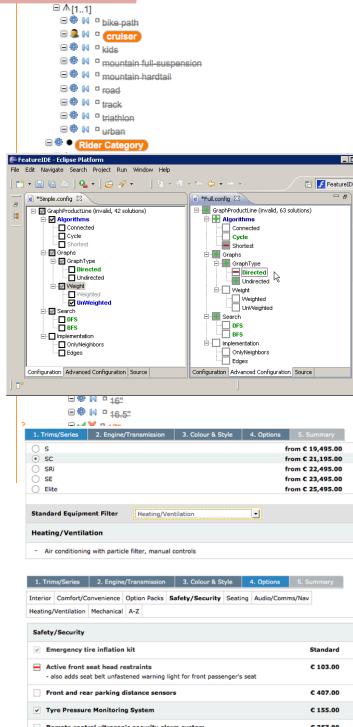
1

GUI metamodel  
(subset of UsiXML)



GUIModel-2-concrete  
Transformation

3



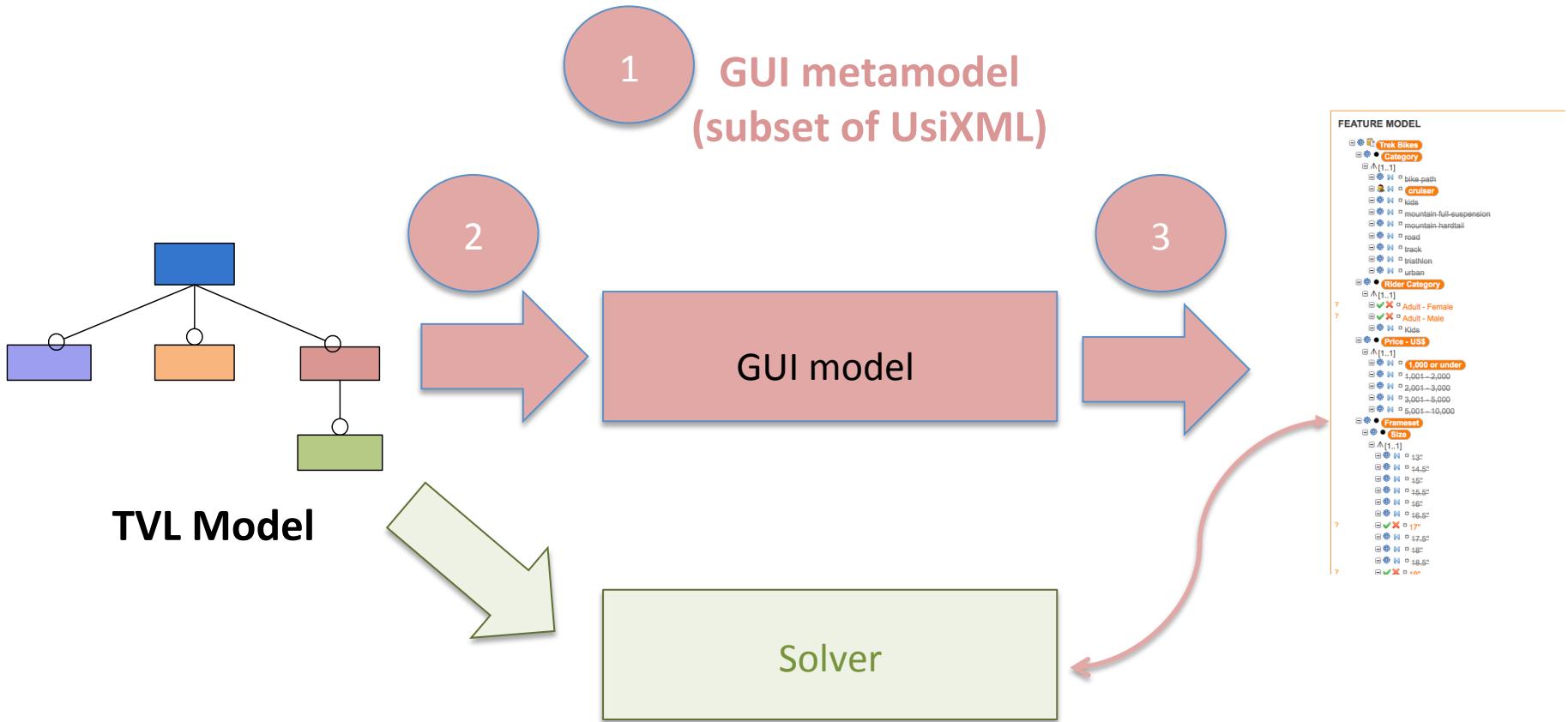
# Running project

- Expected output
  - Configurator artefacts
    - TVL-2-GUIModel transformation
      - “generic rules”
      - API of TVL
    - metamodels/models
      - Analysis of your configurator and your technological space
      - Ideally: subset of UsiXML
      - Ideally(2): merge your efforts
    - GUIModel-2-concrete transformation
      - Connection with the solver
- Presentation
  - “Defense” (20’ + 10’ questions)
    - demo
  - Don’t forget to report on “open issues” or “difficult problems”

# Running project

- Good properties
  - Solution that works
  - Reusable artefacts
  - Customizable solution
  - Close to existing configurator (“re-engineering”)
    - Or even better: “better”
- Be incremental
  - Simple metamodel, simple transformations
  - Working solution ASAP
    - then you can tackle more difficult problems
- Collaborative effort
  - Within the group
    - 3 main steps... but inter-related
  - (optional) Between the groups
    - Identify similarities (metamodel?) / reusable artefacts

# Today



## Ecore metamodel (Eclipse modeling)

Suggestion: all members should work on a first version of the metamodel (MM)

Simple MM first, then one member of the group focusing on UsiXML