# 30. Idea Variation for a Mature Feature Model of the Product

Prof. Dr. Uwe Aßmann

Technische Universität Dresden
Software Engineering Group
Version 2020-0.3, 12.12.20
http://st.inf.tu-dresden.de

1) Variation in Component Trees and Feature Models
2) Systematic Inventive Thinking (SIT) on component trees
3) SCAMPER
4) Raijkar's Hexagon
5) SAMM
6) Scalable Costs

# Business Network of the Day



- https://angel.co/

# Obligatory Literature



- Kwanwoo Lee, Kyo C. Kang, and Jaejoon Lee. Concepts and guidelines of feature modeling for product line software engineering. Lecture Notes in Computer Science, 2319:62--78, 2002. Good overview on feature models
- Alexander Grots, Margarete Pratschke. Design Thinking — Kreativität als Methode. Marketing Review St. Gallen, April 2009, Volume 26, Issue 2, pp 18–23
  - DOI: 10.1007/s11621-009-0027-4
- Drew Boyd (Autor), Jacob Goldenberg. Inside the Box: The Creative Method That Works for Everyone. Profile Books Ltd. 2014.
  - Introduces Systematic Inventive Thinking (SIT)
  - http://www.sitsite.com/method/
  - https://en.wikipedia.org/wiki/Systematic_inventive_thinking

Any good business model (also an MVP)
should be improved by new variants or extensions.

# Other Literature



- Don S. Batory. Feature models, grammars, and propositional formulas. In J. Henk Obbink and Klaus Pohl, editors, Software Product Lines, 9th International Conference, SPLC 2005, Rennes, France, September 26-29, 2005, Proceedings, volume 3714 of Lecture Notes in Computer Science, pages 7--20. Springer, 2005.
  - Explains the relationship of feature models and propositional logic.
- Hans de Bruin and Hans van Vliet. Quality-driven software architecture composition. Journal of Systems and Software, 66(3):269--284, 2003.
  - Introduces feature-solution graphs, the bipartite graph between feature trees and product-component trees.

# Improving a BMC or Developing a New One?



- When a BMC has been graded and assessed, some its fields may need to be improved or ***varied (exchanged)***
  - Variation yields “greener” canvases
- A *red BMC* or a *red VPC* (failing the assessment) should be changed
  - If there is no successful sticky and viral MVV, there is a problem

**Point of Pivot:** [Blank] Sometimes, this does not help and the BMC must be thrown away, and a **plan B** has to be found, another BMC.

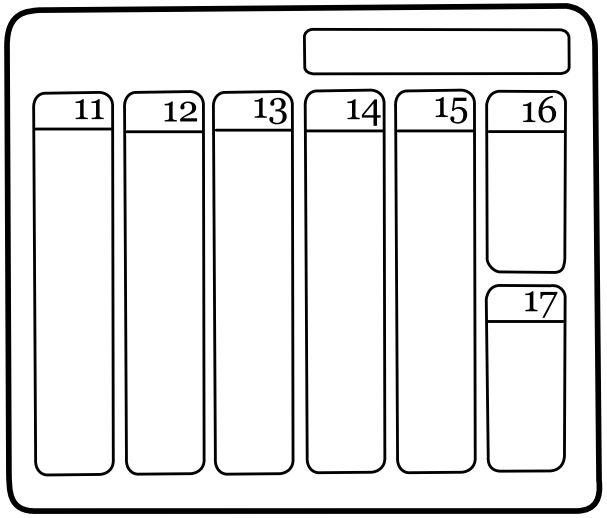
- This chapter introduces systematic ways to change (vary, exchange) the fields of
  - Value proposition
  - Key resources
  - Customer segments

# 30.1. Canvases for Idea Generation

# Shortcomings of Lean Startup from the Viewpoint of Software Product-Line Engineering



No support for consistent modeling of product lines (no support for feature modeling and feature variation)




No support for canvas modeling (composition and engineering)

No support for staged feature configuration with suppliers

No support for grading and metrics

# Remember the Value of the Variation-Based Business Model (Software Product Lines)



„**Software product lines** represent perhaps the most exciting paradigm shift in software development since the advent of the high-level programming languages. Nowhere else in software engineering have we seen such breathtaking improvements in cost, quality, time to market, and developer productivity, often registering in the order-of-magnitude range.“

„At the Software Engineering institute, we have recorded case study after case study of companies succeeding in one market area with a product line approach, and then taking their production capability to a nearby, under-exploited area of the market, and quickly rising to market dominance in that area as well. And why not? **If you can outperform your competitors by order-of-magnitude levels**, it's hard to imagine what could you keep from becoming a market leader.“

Paul Clements, SEI, in „Software Product Lines in Action“, Springer-Verlag.

# Techniques for Idea Generation and Their Canvases



- A canvas can be used for scaling a business

## BMC Variation

- For systematic variation of the fields of BMC
- with 4 structured operations of the book „Business Model Generation“

## S.I.T. Canvas (Inside-the-box canvas)

- For systematic variation with Systematic Inventive Thinking (S.I.T.)

## SCAMMPERR Canvas

- Structured process for idea variation

## Hexagon Variation Canvas

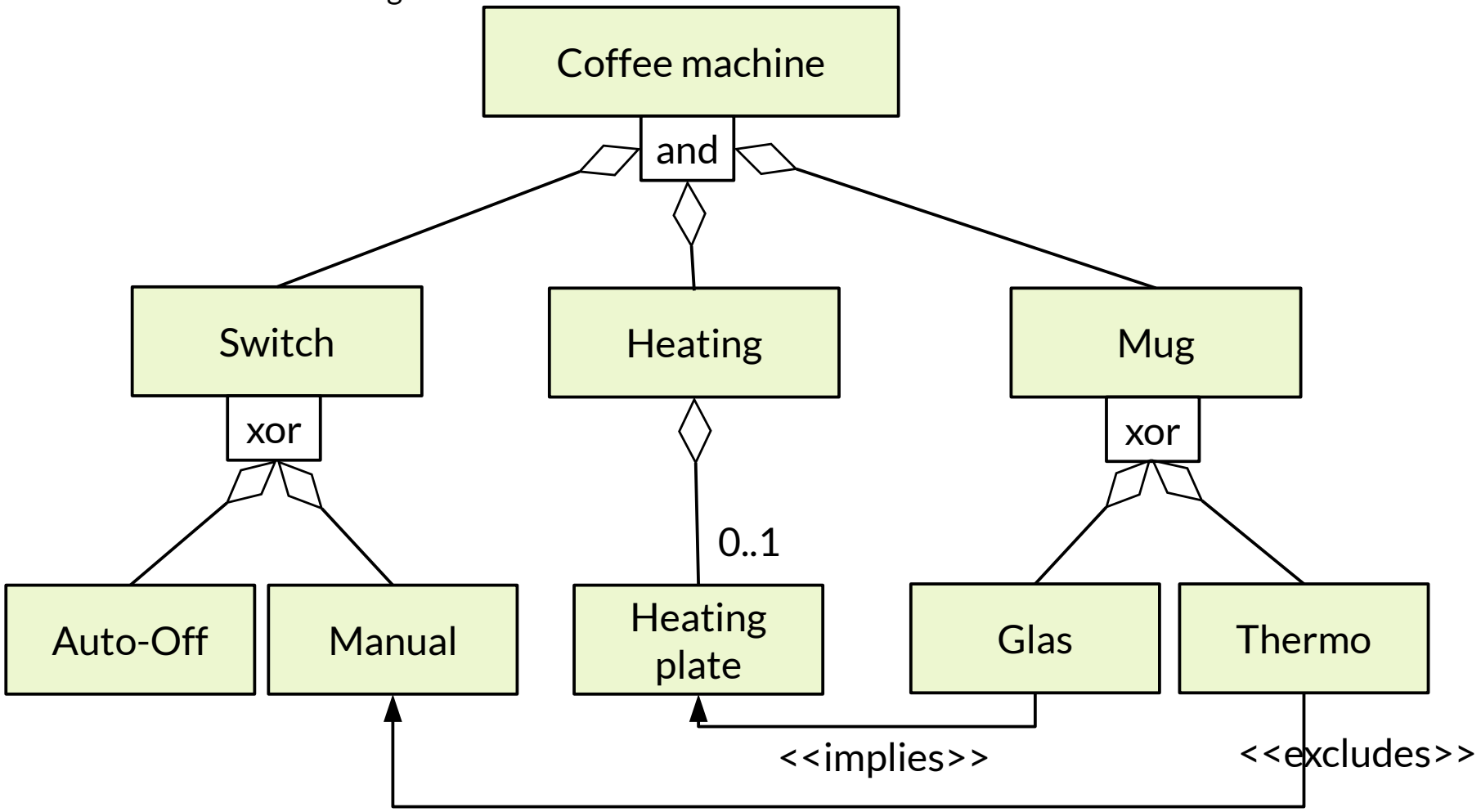
- Structured process for variations
- Priorization of variants
- For scaling

# 30.1.1. Product Component Trees and Feature Trees

# Product Component Tree



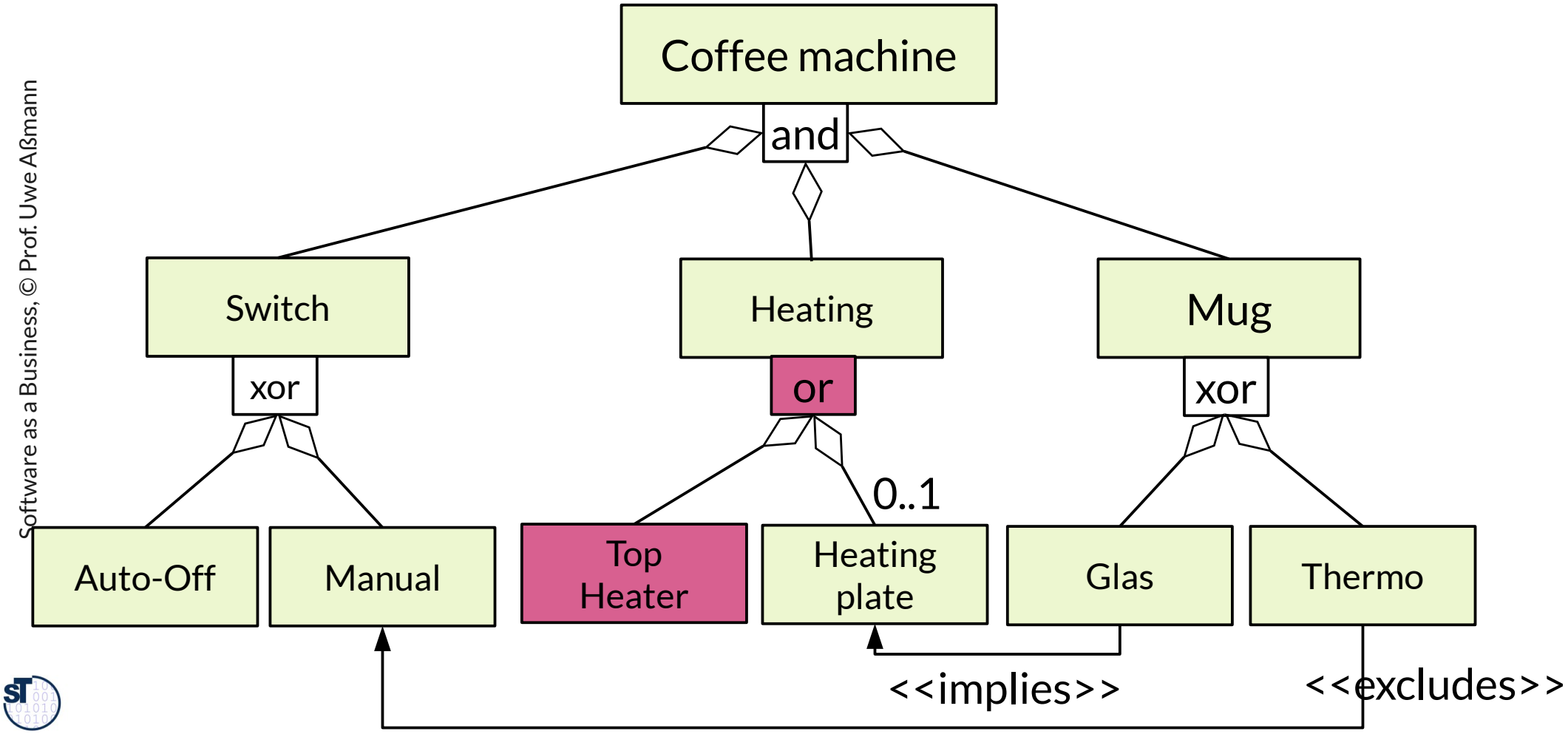
- A **component tree** is a and/or link tree of the components of a product, with options, inclusion and exclusion constraints.
  - It describes a combinatorial variant space of *components* and can be mapped to propositional logic
  - Product Component Trees generalize Product Breakdown Structure (PBS) from Course Softwaremanagement



# Varianting Means to Add Alternatives to a Product Component Tree



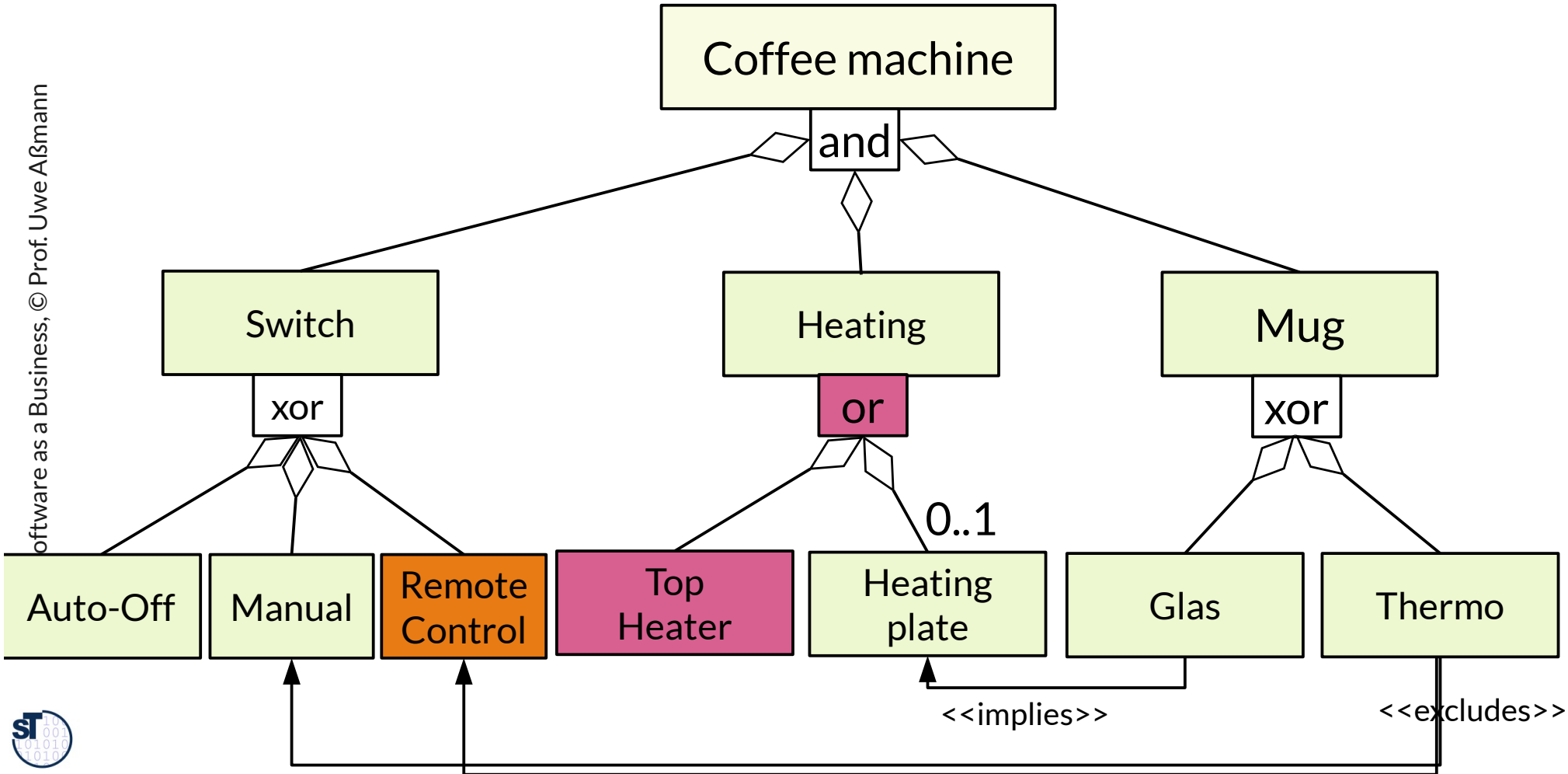
- Variation adds
  - new alternatives to an OR or XOR node
  - New OR or XOR nodes to AND nodes



# Varianting Means to Add Alternatives to a Product Component Tree (2)



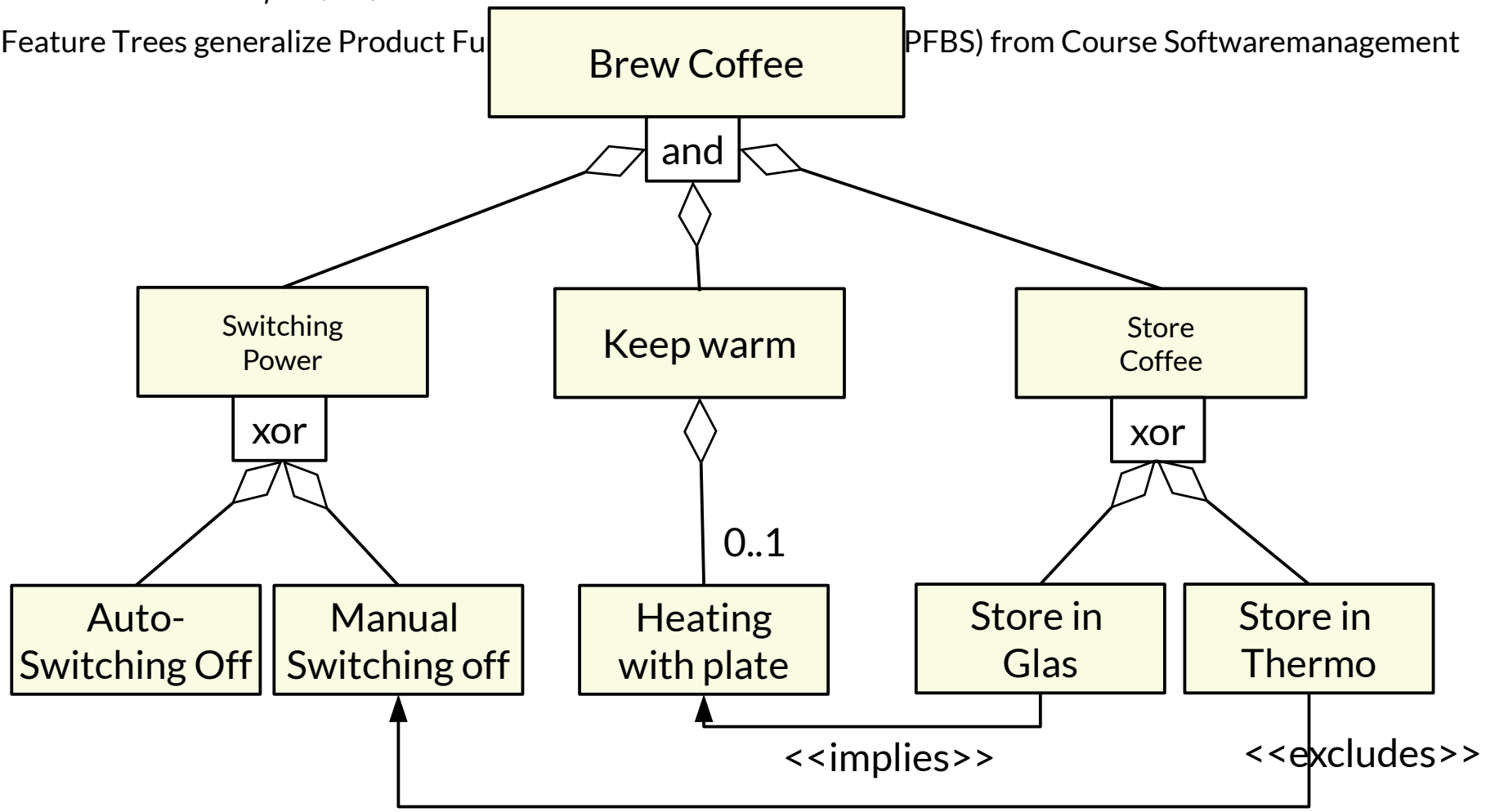
- Step by step, new components (for new features) can be added



# Feature Tree (Feature Model)



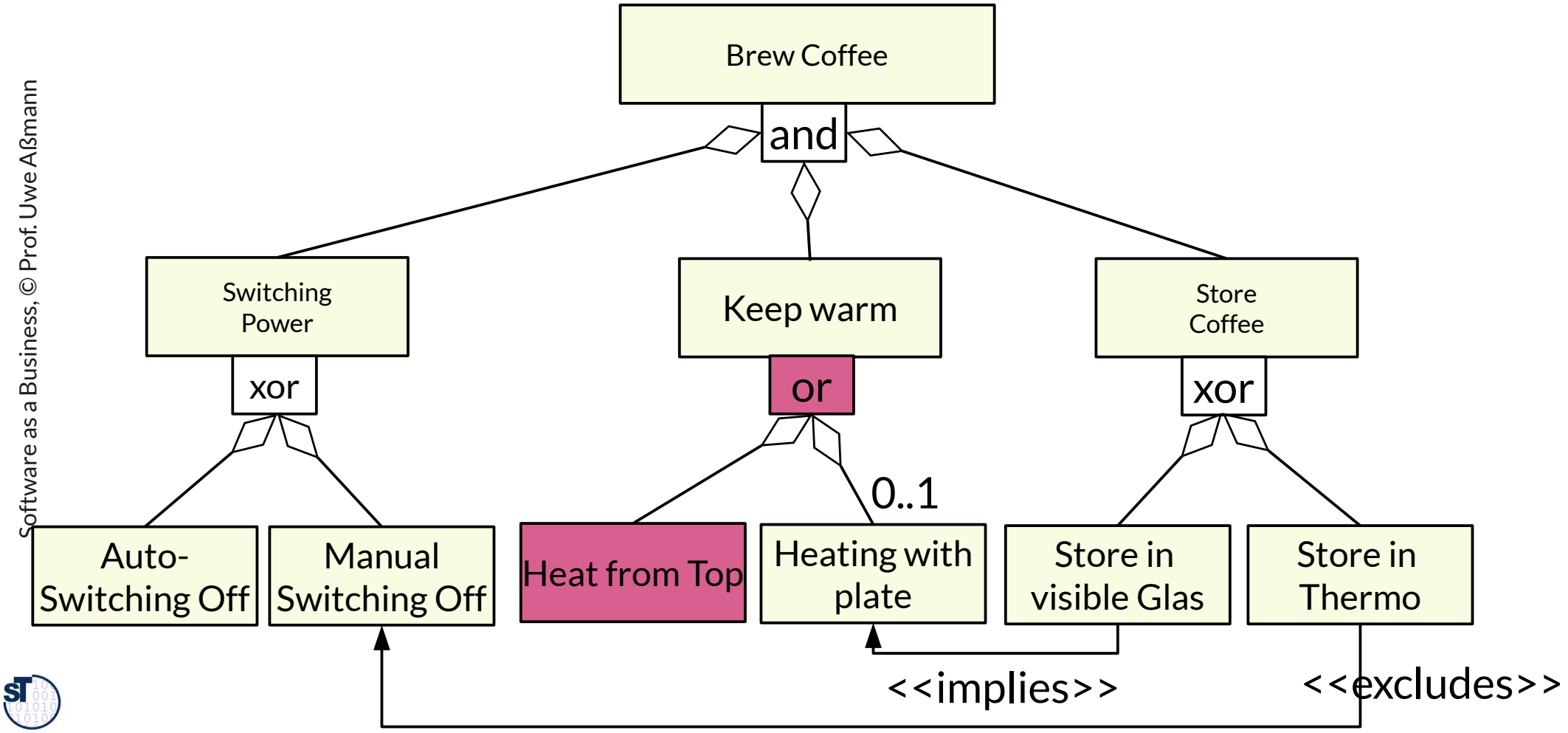
- A **feature tree (feature model)** is a and/or link tree of *functions (features)* with options, inclusion and exclusion constraints.
  - Functional decomposition
  - It describes a combinatorial variant space of *functions* and can be mapped to propositional logic over *functions*
- Feature Trees generalize Product Fu PFBS) from Course Softwaremanagement



# Varianting Means to Add Alternatives to a Feature Model



- Variation adds new **feature alternatives** to an OR or XOR node
  - New OR or XOR **feature nodes** to AND nodes
- Attention: feature trees are not component trees!



# Idea and Feature Variation with Feature Trees



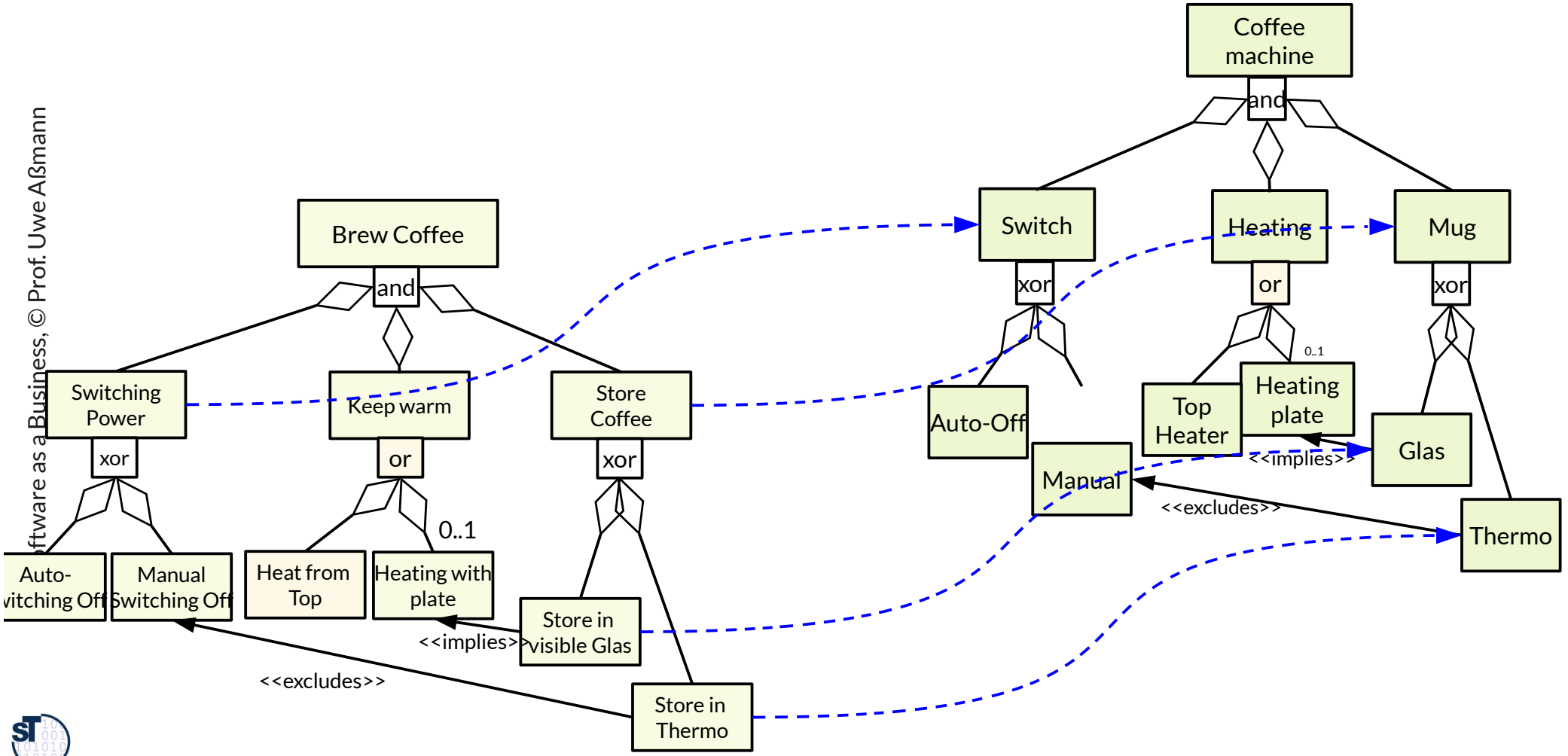
- Business model:
  - Product-oriented
  - Product-line-oriented
  - Software ecosystem: Features are distributed into apps on a software platform
- *Exercise: How to extend the features of a coffee machine?*

# Feature-Solution Bigraphs



A ***feature-solution (bi-)graph*** maps a feature tree to a product tree via a *feature-solution mapping*

- Bigraphs contain two or more graphs (dimensions) linked by a *link graph*



# Warning - The Scale Trap



- Many companies start *without* feature tree, product component tree, and feature-solution graph
  - -> They have a hard time finding and developing the components of the product, as well as their integration
- After 2 years, when they want to scale, they change to a product-line business model
- Feature trees and product component trees are *indispensable* for the management of a product line
- Then, the feature tree has to be *reconstructed*

**Law of scaling:**
If you want to scale:
Maintain a feature tree and a product component tree, as well as their feature-solution mapping
from the MVV on

# Next Pitches



On Fri, Dec 18, 2020, there will be the „landing page and smoke video“ pitch.

Please also prepare a component tree or feature tree to motivate how you can scale.

On Fri, January 22, 2020, there will be the first MVP pitch.

Please also prepare a component tree or feature tree to motivate how you can scale.

On Mon, Febuary 1, 2021, 16:40, there will be the „Dungeon of Dragons“.

# 30.2. Change-Driven Invention of new Products Variability-Based Design

- For scaling, it is important to develop *alternatives* and *variants*

- Drew Boyd, Jacob Goldenberg. Inside the Box. Why the best business innovations are right in front of you. Profile Books, London, 2013
- http://en.wikipedia.org/wiki/Systematic_inventive_thinking
- http://en.wikipedia.org/wiki/Unified_structured_inventive_thinking

# 30.2.1 Business Model Development with 4-field Portfolio of BMG, as a Matrix Analysis

[BMG p 231]

# Matrix Analysis with 7W



- A Matrix Analysis combines two dimensions, for one canvas a set of questions or concepts
- With a matrix analysis, we create new ideas

| Questions 7W | Key Partners | Key Activities | Key Resources | Costs | Value Propositions | Customer relationships | Channels | Customer Segments | Revenues |
|---|---|---|---|---|---|---|---|---|---|
| Who? | | | | | | | | | |
| What? | | | | | | | | | |
| When? | | | | | | | | | |
| Where? | | | | | | | | | |
| Why? | | | | | | | | | |
| What for? | | | | | | | | | |
| How? | | | | | | | | | |
| | | | | | | | | | |

# 4-Actions Variation Framework of BMG



- The BMG book presents 4 operators for new ideas:
  - Eliminate
  - Reduce
  - Augment
  - Create

- To model the influence of these dimensions of the BMC, we have to span up a matrix with 4x9 elements (matrix analysis)

# Matrix Analysis for 4-Actions-BMC



- For this aspect-oriented matrix analysis for the BMC, create a table (matrix) of 4-actions and BMC, brainstorm on the crossproduct

| | Key Partners | Key Activities | Key Resources | Costs | Value Propositions | Customer relationships | Channels | Customer Segments | Revenues |
|---|---|---|---|---|---|---|---|---|---|
| Eliminate | | | | | | | | | |
| Reduce | | | | | | | | | |
| Augment | | | | | | | | | |
| Create | | | | | | | | | |



[BMG p.233ff]

# Matrix Analysis for BeNiSiLo-BMC



- For this aspect-oriented canvas analysis on AUGMENT, create a table (matrix), brainstorm on the crossproduct
- The “operations dimension” is BeNiSiLo, a quality-oriented set of improvement operations

| Augment | Key Partners | Key Activities | Key Resources | Costs | Value Propositions | Customer relationships | Channels | Customer Segments | Revenues |
|---|---|---|---|---|---|---|---|---|---|
| Better | | | | | | | | | |
| Nicer | | | | | | | | | |
| Simpler | | | | | | | | | |
| Longer Lasting | | | | | | | | | |

# 30.2.3. Variability-Based Business with Systematic Inventive Thinking (SIT)

# SIT operates on Component Trees (Component Tree Algebra)



[https://en.wikipedia.org/wiki/Systematic_inventive_thinking]

- SMUDAD-operations on products and their component trees
- Also possible on feature trees

| | | |
|---|---|---|
| S | Subtract (Eliminate) | Remove, subtract components, reduce to core („Steve Jobs pattern“) |
| M | Multiply | Add another component, potentially different component to the product |
| U | Unify tasks | Find a new task for a component so that it can deliver two tasks |
| D | Divide | Re-group the components of the product into subgroups and form a new product (product out-lining). A first step to a product-line oriented business model |
| AD | Attribute dependency | Remove or create dependencies between parameters of components |