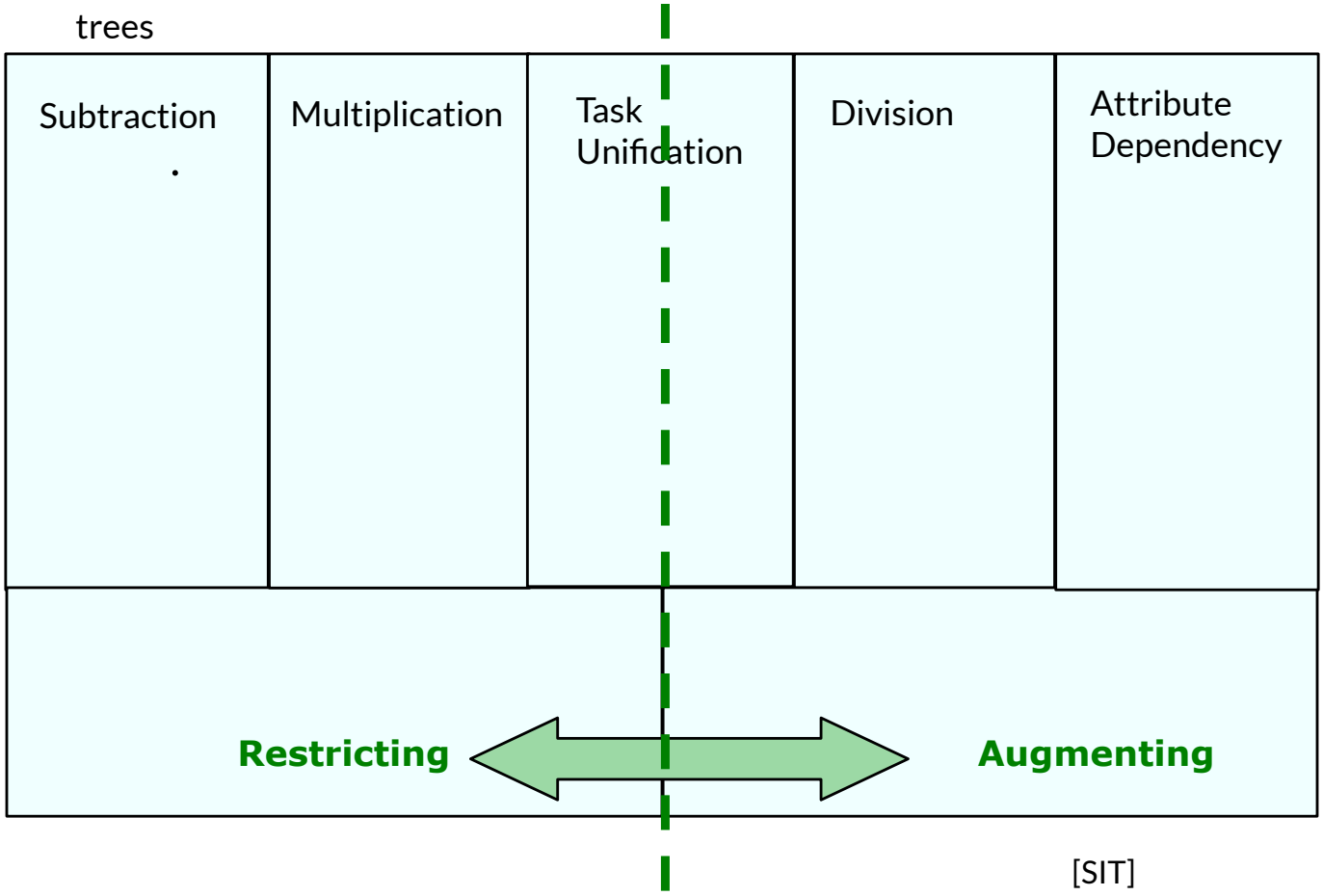
SIT Canvas with SMUDAD Operations

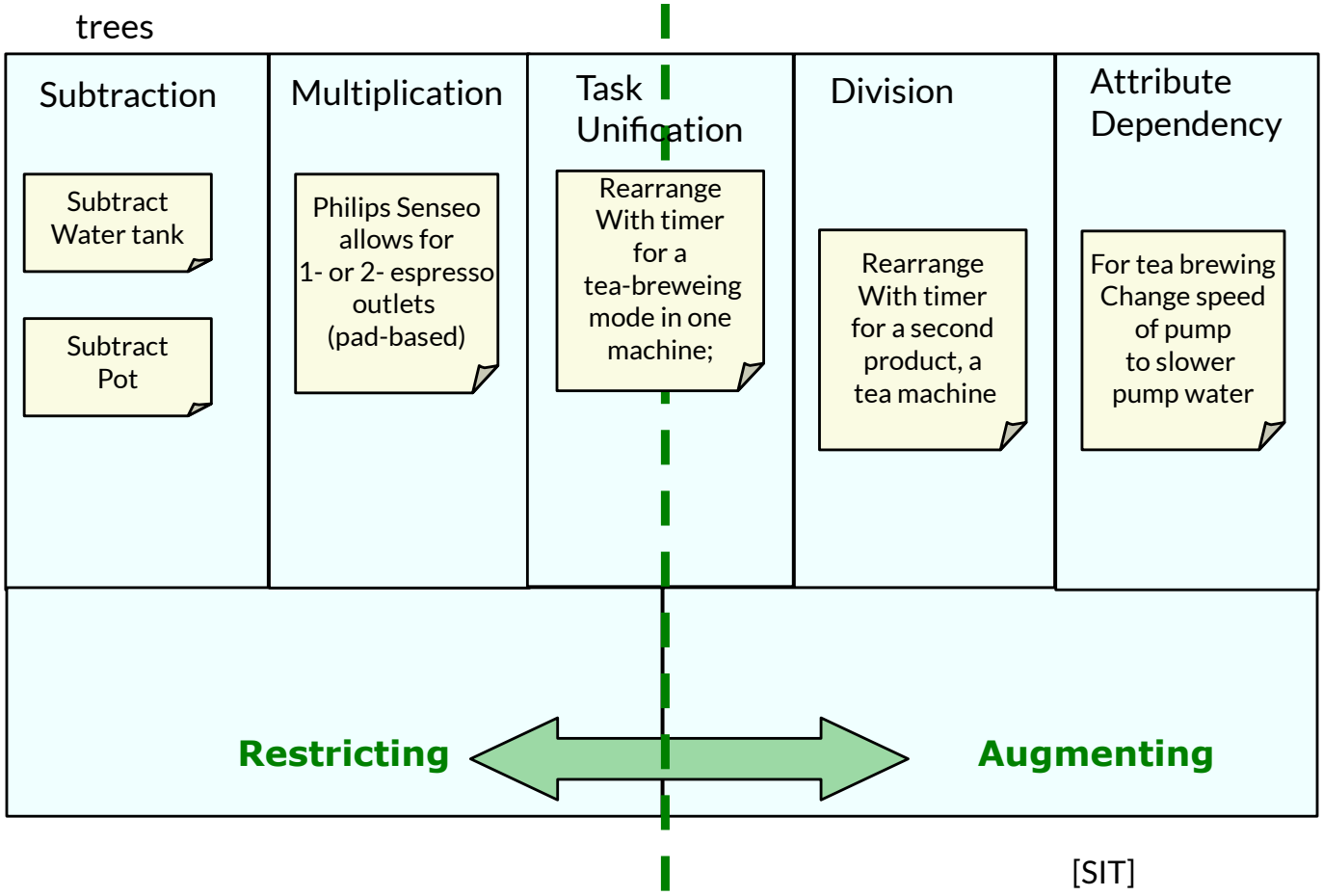
- SIT Canvas is based on simple modification operations of existing product component trees



[SIT]

Example: SIT Canvas on Coffee Machine

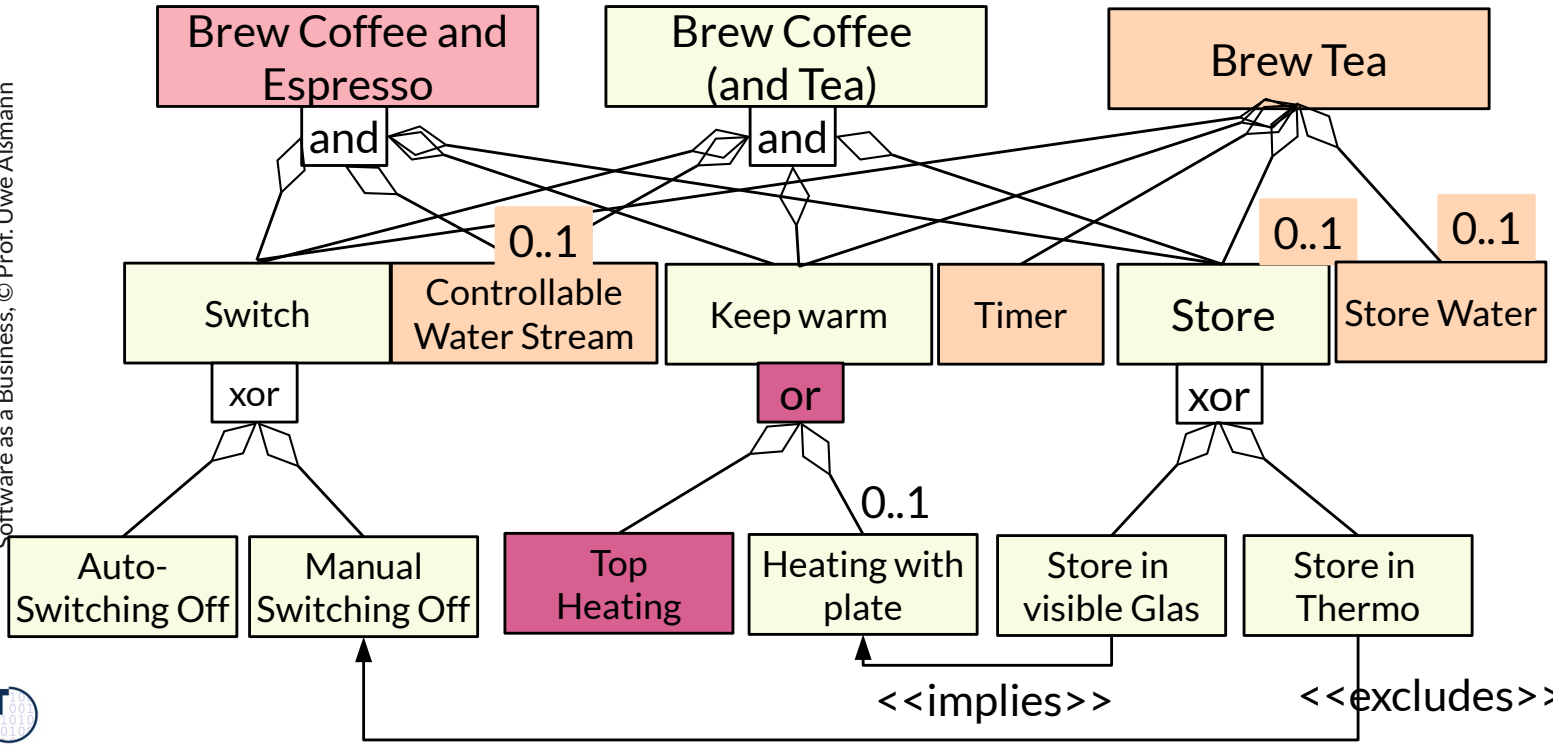
- SIT Canvas is based on simple modification operations of existing product component trees



[SIT]

Extended Feature Model (Multihierarchical Feature Model of Product Line)

- Variation adds 2 new products (Tea machine, coffee+pad-espresso machine)
- CoffeeMachine with enriched feature set
- Feature model may become too complex → refactoring necessary



SIT thinking
Subtraction Technique

- Subtracting components from the component set of a product
- Implied: removing features from the feature set of a product
    - Make it simpler and easier to use
    - Reduce costs

Examples:

- Steve Jobs was great in subtractions
    - Ipod with very few knobs
    - Ipad has no keyboard (compare to Microsoft Surface)
        - No USB
        - No CD/DVD
        -

Division (Decomposition) Technique

- A CD, radio, cassette player all contain amplifyers.
- An *integrated music center* contains a CD, radio, cassette player and amplifyer.
  - One amplifier provides amplification for every other device.
  - Function is *divided*
- A *modular music center* is composed of components that can be replaced
  - Function is divided and replacable

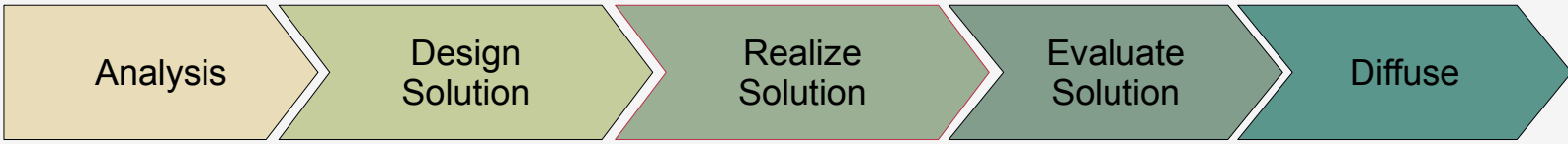
Matrix Analysis SITxBMC

- For this aspect-oriented canvas analysis, create a table (matrix), brainstorm on the crossproduct

	Key Partners	Key Activities	Key Resources	Costs	Value Propositions	Customer relationships	Channels	Customer Segments	Revenues
Subtraction									
Task Unification									
Multiplication									
Division									
Attribute dependency									

30.3 SCAMPER Idea Variation

- SCAMPER is a Solution process (see course ASICS)



http://de.wikipedia.org/wiki/SCAMPER

[Bob Eberle 1997]

- SCAMPER is a variation technique with 6 algebraic variation operators
- Derived from OSBORN checklist
- Kilbride's SCAMMPERR (SCAMPER+) adds two variation operations

S	Substitute (Vary)	Substitute some parts of the solution, resources, channels etc.
C	Combine	Combine partial solution elements to a more complete solution
A	Adapt	Change the solution or function
M	Modify	Scale, change an attribute of the solution
M	*Magnify*	*Change the size of the solution*
P	Put	Put to (find) another use
E	Eliminate (Subtract)	Remove, subtract, reduce to core
R	Reverse	Invert order
R	*Rearrange*	*Change order*

Ex.: SCAMMPERR with Sensor-Based Diapers

[Bob Eberle 1997]

- Remember the water-sensor-based diapers...

S	Substitute (Vary)	Substitute a part: Substute cable of sensor against wireless
C	Combine	Combine partial solution elements to a more complete solution: Second app to do social community analysis, taking the analytics of other parents into account
A	Adapt	Change the solution or function: Do a sensor-based diapers for elderly and handicapped people
M	Modify	Scale, change an attribute of the solution:
M	Magnify	Change the size of the solution: Make the wireless sensor smaller to be taken into the bladder; and use it for incontinent people
P	Put	Put to (find) another use
E	Eliminate	Remove, subtract, reduce to core: Let the sensor ring – no app
R	Reverse	Invert order
R	*Rearrange*	*Change order*

Matrix Analysis SCAMPERxBMC

- For this aspect-oriented canvas analysis, create a table (matrix), brainstorm on the crossproduct
- Exercise: do the same for VPC and PainCanvas

	Key Partners	Key Activities	Key Resources	Costs	Value Propositions	Customer relationships	Channels	Customer Segments	Revenues
Subtraction									
Combine									
Adapt									
Magnify/ Modify									
Put									
Rearrange/ Reverse									

Matrix Analysis SCAMPERxVPC

- For this aspect-oriented canvas analysis, create a table (matrix), brainstorm on the crossproduct
- Exercise: do the same for VPC and PainCanvas

	Customer Tasks	Gains	Pains	Pain Killers	Gain creators	Advantages	Features
Subtraction							
Combine							
Adapt							
Magnify/ Modify							
Put							
Rearrange/ Reverse							

30.4. Variability-Based Business with Rajkar's Idea Hexagon

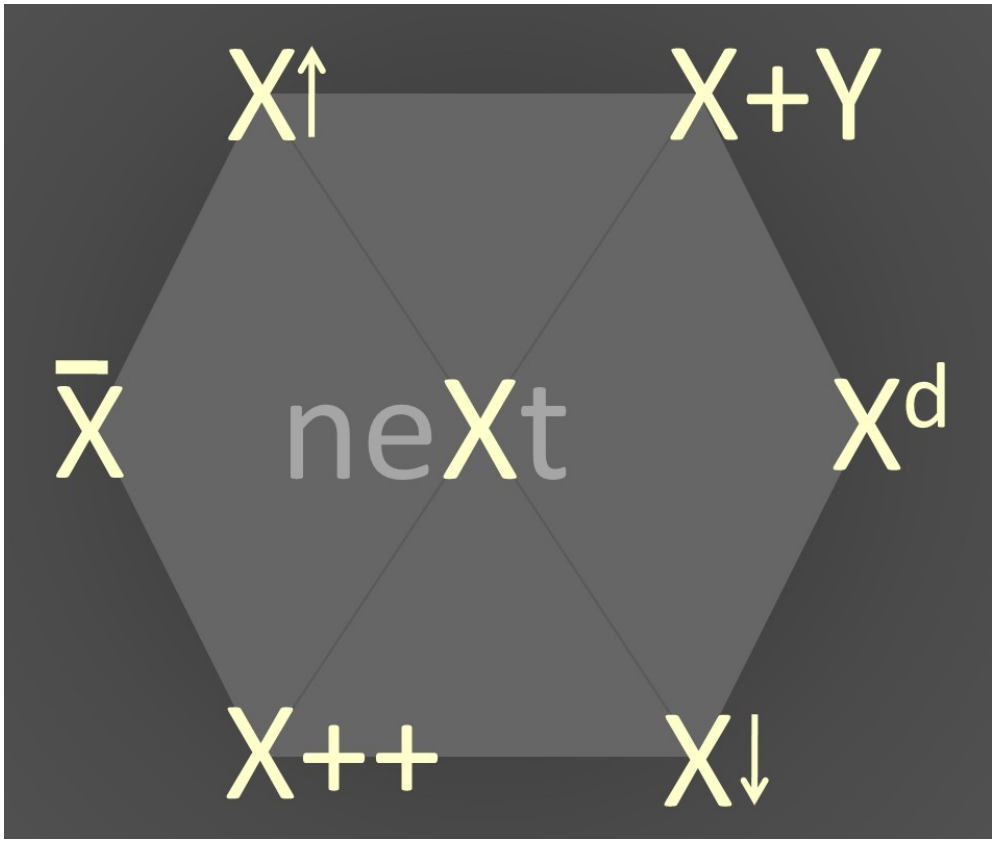
Slideshare Lecture of Rajkar / MIT.

The 6 Operations of Rajdar's Idea Hexagon

 [http://de.slideshare.net/cameraculture/raskar-ideahexagonapr2010]

- 6 Operations (DROHNF) to get new ideas (not only for products, also for technology)

D	Dimensional extension	Add another dimension to the idea
R	Restricting adjective	Add a new constraining adjective to the solution
O	Opposite	Do exactly the opposite
H	Find a hammer for a nail (abstracting, frameworking)	Search a new generic idea for an application; a new solution for a problem
N	Search for a new nail for the hammer (re-concretizing, framework re-instantiating)	Search a new application for a generic idea; a new problem for a solution
F	Fusion	Fuse dissimilar ideas into one idea



By Mituser2000 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=74796337

Idea Hexagon x VPC

- The operations are important for Product Line Engineering:
  - Dimensional extension (creates Product Matrices)
  - Hammer (creates frameworks)
  - Nail (instantiate frameworks)

	Customer Tasks	Gains	Pains	Pain Killers	Gain creators	Advantages	Features
Dimensional extension							
Restricting							
Opposite							
Find Hammer							
Find Nail							
Fuse							

30.5 SAMM

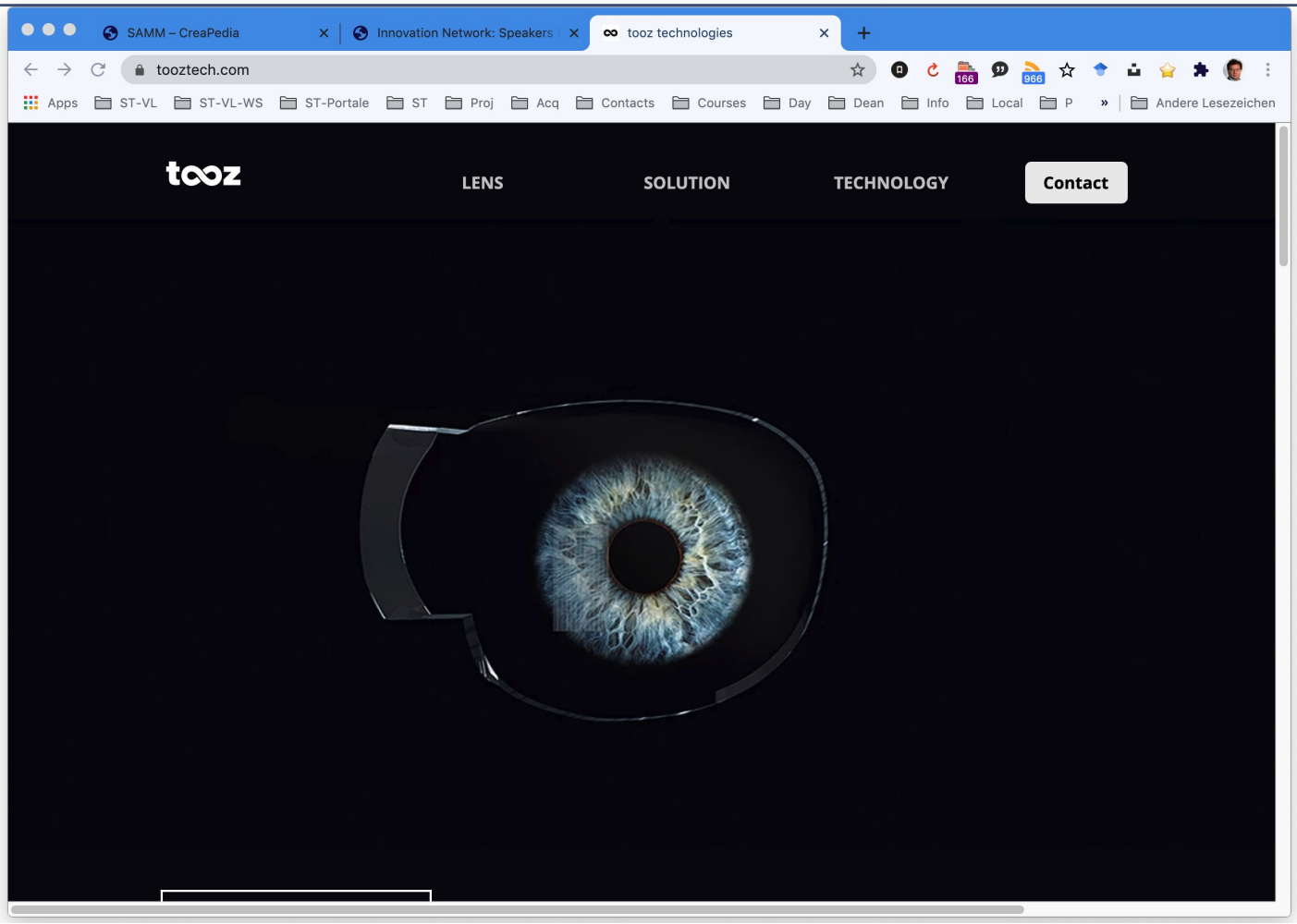
SAMM http://www.creapedia.com/w/index.php5/SAMM
als SCAMPER+ für Aktivitäten

Ex.: SCAMPERR with Sensor-Based Diapers

- Think about the steps of a process

S	Substitute (Vary)	Substitute a process step
C	Combine	Combine several process steps to a macro-step
A	Accellerate	Faster...
M	Modify	Scale, change an attribute of the solution:
P	Prioritize	Prioritize process steps in importance
E	Eliminate	Remove a process step
R	Reverse	Invert order of process
R	Rearrange	Change order, parallelize

Tooz iGlasses



Ex.: SCAMPERR with Sensor-Based Diapers

- Refactor the steps of controlling the humidity of your child's diapers

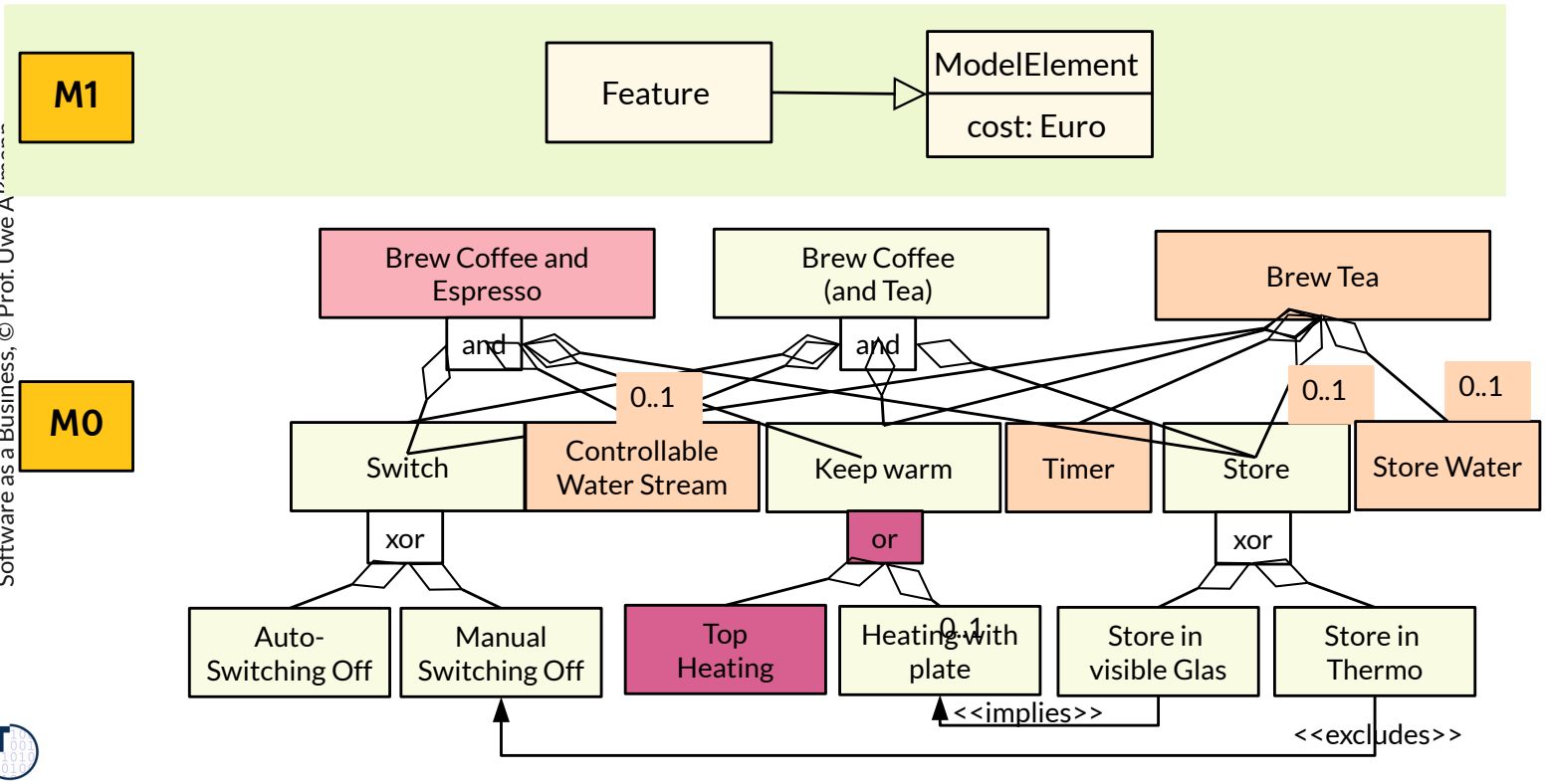
S	Substitute (Vary)	Substitute a process step: control humidity by app
C	Combine	Combine several process steps to a macro-step: check the 14-days analytics
A	Accellerate	Faster... Use a tooz glasses to blend in the status of your child's diapers into your eye
M	Modify	Scale, change an attribute of the solution: Vary humidity level (dry, semi-dry, wet, real-wet)
P	Prioritize	Prioritize process steps in importance: weigh humidity warning vs. humidity ignorance in groups
E	Eliminate	Remove a process step: eliminate manual intervention
R	Reverse	Invert order of process: not possible
R	Rearrange	Change order, parallelize. Let the app chose whether father or mother changes diapers

30.6. Incremental, Scalable Costs

The left side of the BMC talks about costs. How can we make them incremental and scalable?

Costs of Features in a Feature Model (of Product Line)

- Every feature node may have a ***cost attribute***
- Def.: The ***cost of a feature*** is the sum of the costs on the feature path
- Def.: The ***cost of a product*** with a set of features is the sum of all feature costs
- Def.: The ***cost of a product line*** is the cost of all nodes in its feature model

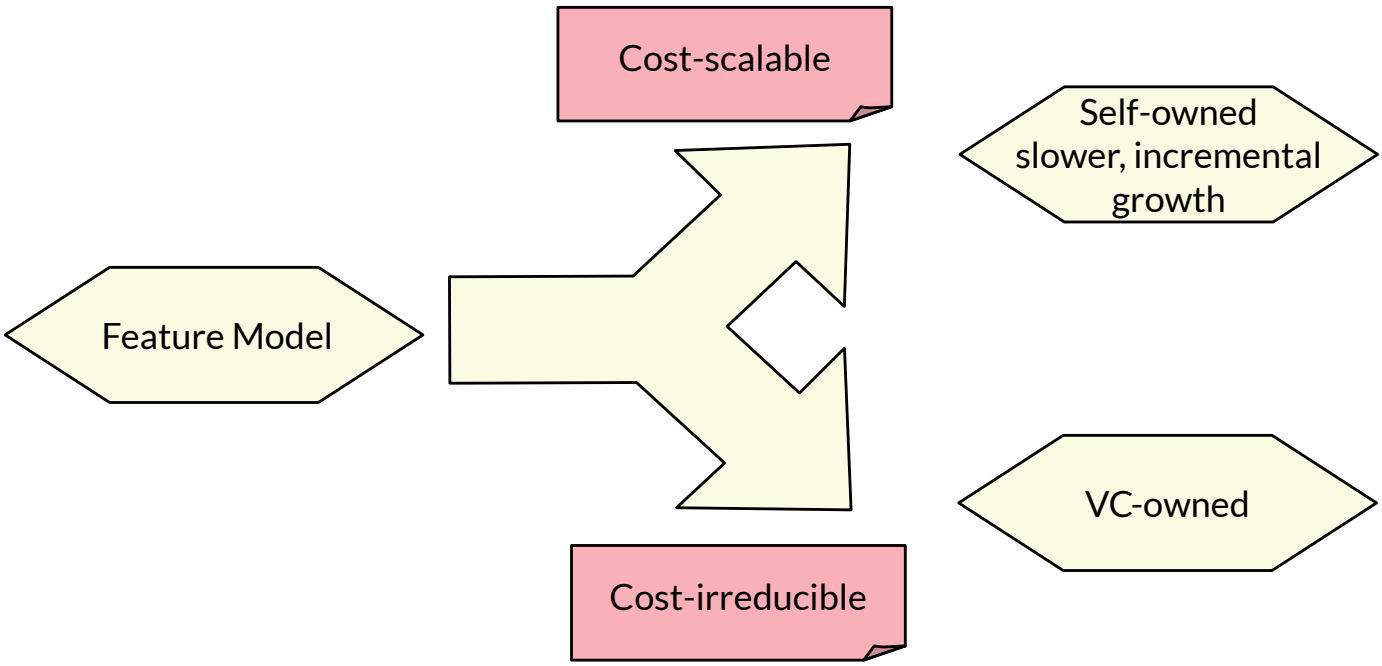


Cost Scalability and Venture Capital

- A company has a problem with an MVP if the cost of one feature is too high.
- It is better to have feature models with low costs per feature, because then new features can be added easily to the MVP
- Def.: A ***cost-scalable feature model*** is a feature model with low costs for every feature (i.e., all nodes).
- Def.: A ***cost-irreducible feature*** is a feature with high costs the company cannot bear alone, but needs capital partner

Law of Venture Capital (VC):
Startups with a cost-scalable feature model do not need VC.
Startups with a cost-irreducible feature need VC.

Two Ways for Building a Company



Ultimate Cost Advantage of an SPL (SPL-UCoA)

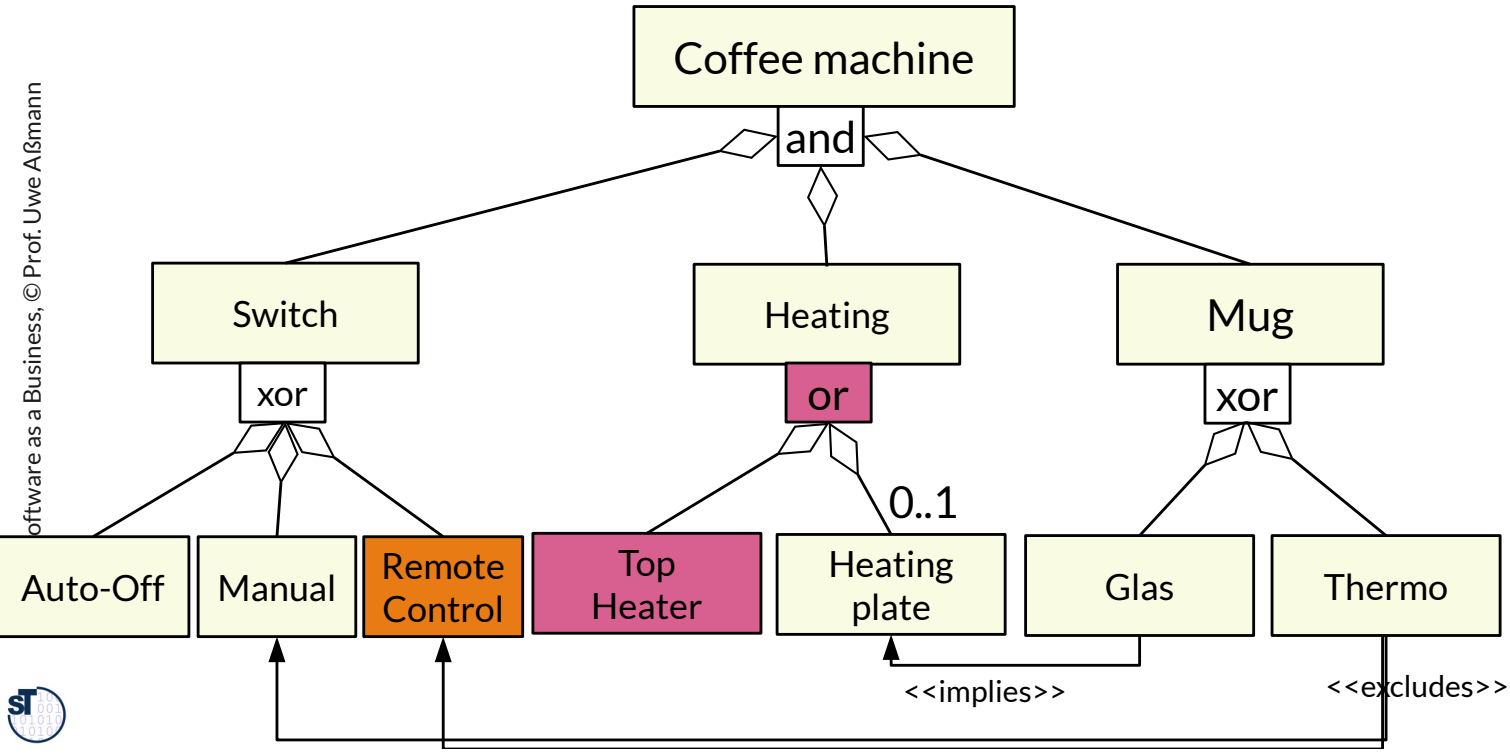
- Ultimate Cost Advantage (UCoA) is a UCA in the cost dimention.
- Make sure that all features in an SPL are extendable (scalable), and share subfeatures
- Then all products in the SPL add their value for the customer (and customer buys more products).

Law of SPL-UCoA:

Scale the products, but limit the costs by reuse of components and features.

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

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



No Idea for Scaling your Business?

- From an MVP, use an idea variation technique to arrive at:
  - Products with more features
  - Products with parameters
  - Feature and component outlining lead to product lines
  - Software ecosystems result if you allow third parties to do the idea variation, i.e., program their own apps
- Plan the scaling early on, but implement step by step
  - The first customers have to finance the next ones
  - Keep the IPR inhouse, only sell non-exclusive licenses to customers

The End

- What is the difference of a component tree and a feature tree?
- Why does the MVP focus on minimal viable *features* instead of minimal viable *components?*
- How do you extend a feature model of an MVFS with more alternative features? Give an overview of the major process steps.
- Explain Raijdar's Idea Hexagon and how to use it to generate new ideas.
- How do you use SCAMPER to get new product features?
- Explain the difference of SCAMPER and S.I.T.
- Suppose you have identified a MVFS, how to find more features?
- Why is it important to cross the 4 BMG operations with the BMC?
- Why is SAMM important for customer touchpoint analysis?