Technische Universität Braunschweig

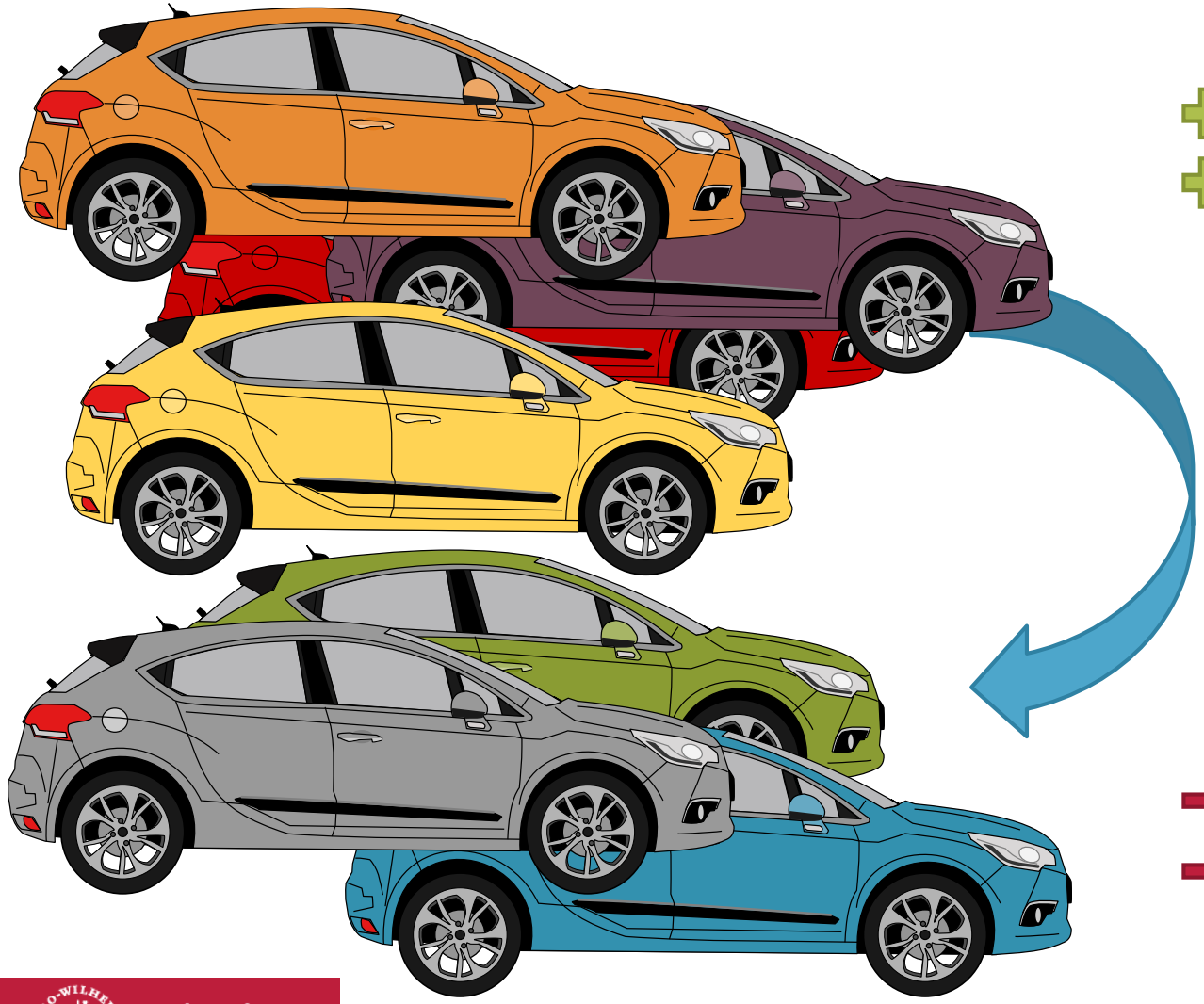
ISF

Variability Mining for Extractive Software Product Line Engineering of Block-Based Modeling Languages

David Wille – October 26th, 2016

Institute for Software Engineering and Automotive Informatics, TU Braunschweig, Germany

Motivation



- ➕ easily creates new variants
- ➕ saves money in short-term

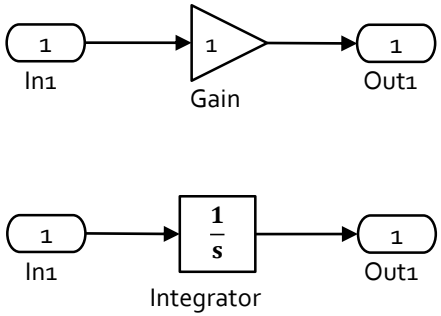
clone-and-own

- ➖ errors are propagated
- ➖ time consuming maintenance

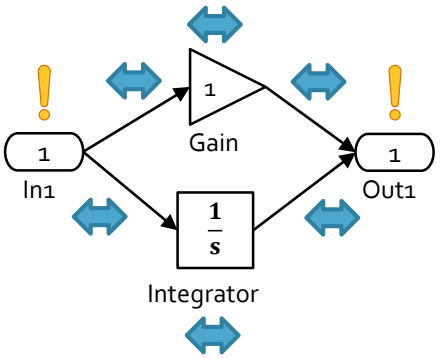
Part I: Variability Mining

Custom-Tailored Variability Mining for Block-Based Languages

Motivation – Variability Mining



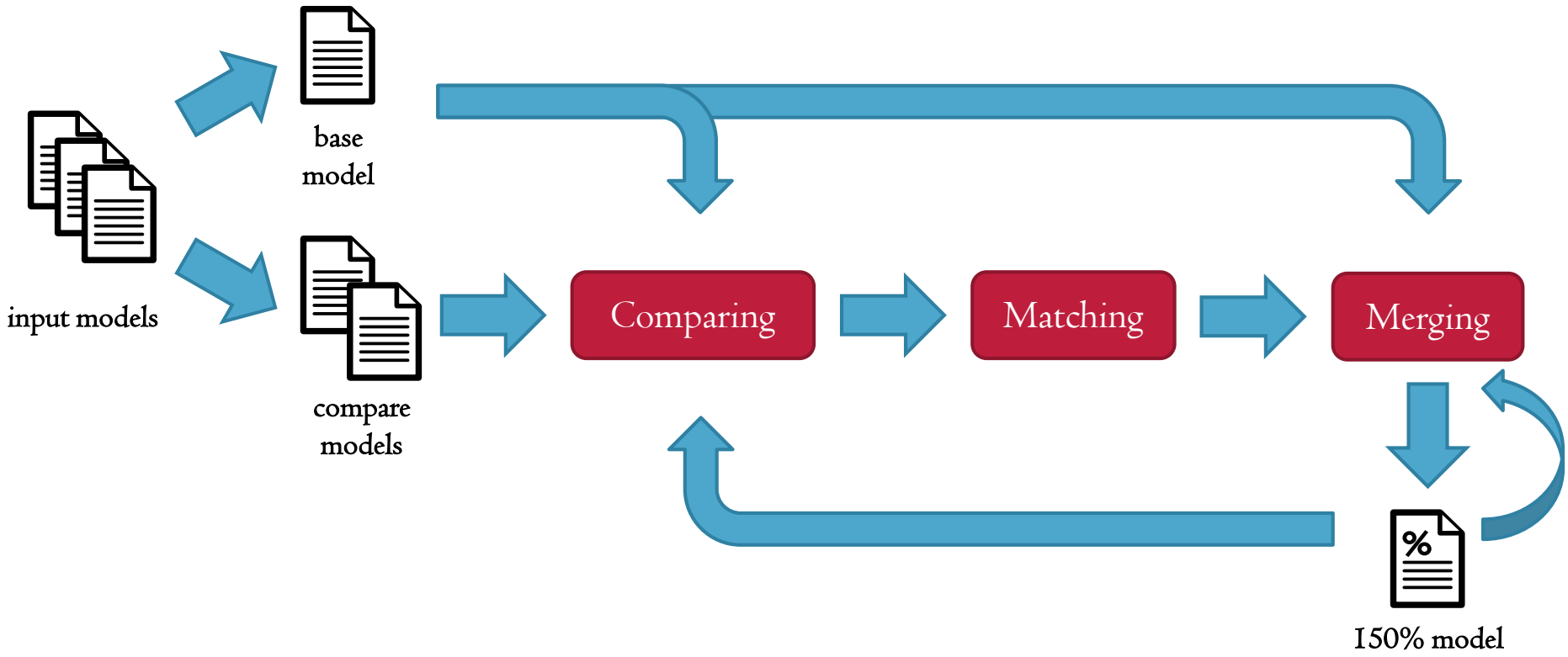
Variability Mining



Motivation – Generic Approach

MATLAB/Simulink	→	Variability Mining for MATLAB/Simulink
State Charts	?	Variability Mining for State Charts
UML Class Diagrams	?	Variability Mining for UML Class Diagrams
System Architectures	?	Variability Mining for System Architectures
…	?	Variability Mining for …

Variability Mining – Workflow



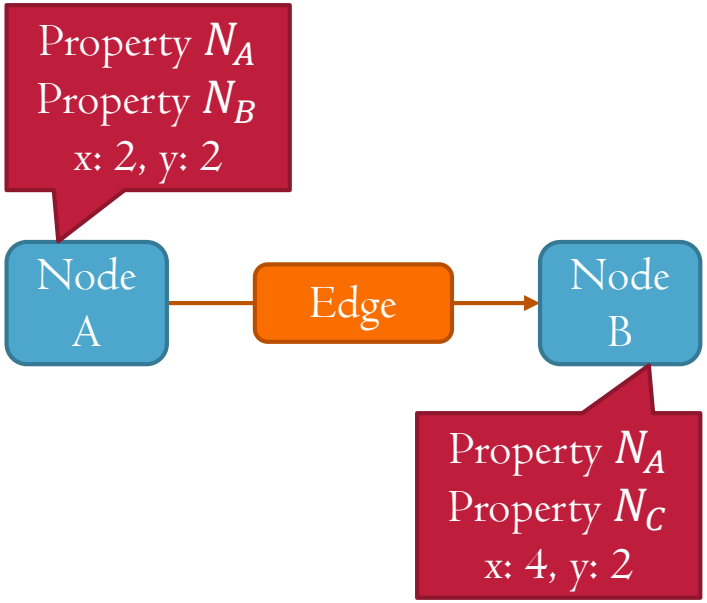
Overview


Analyze the Block-Based Language
Build a Meta-Model
Define a Metric
Adapt the Existing Algorithms

Analyze the Block-Based Language

Relevant Elements

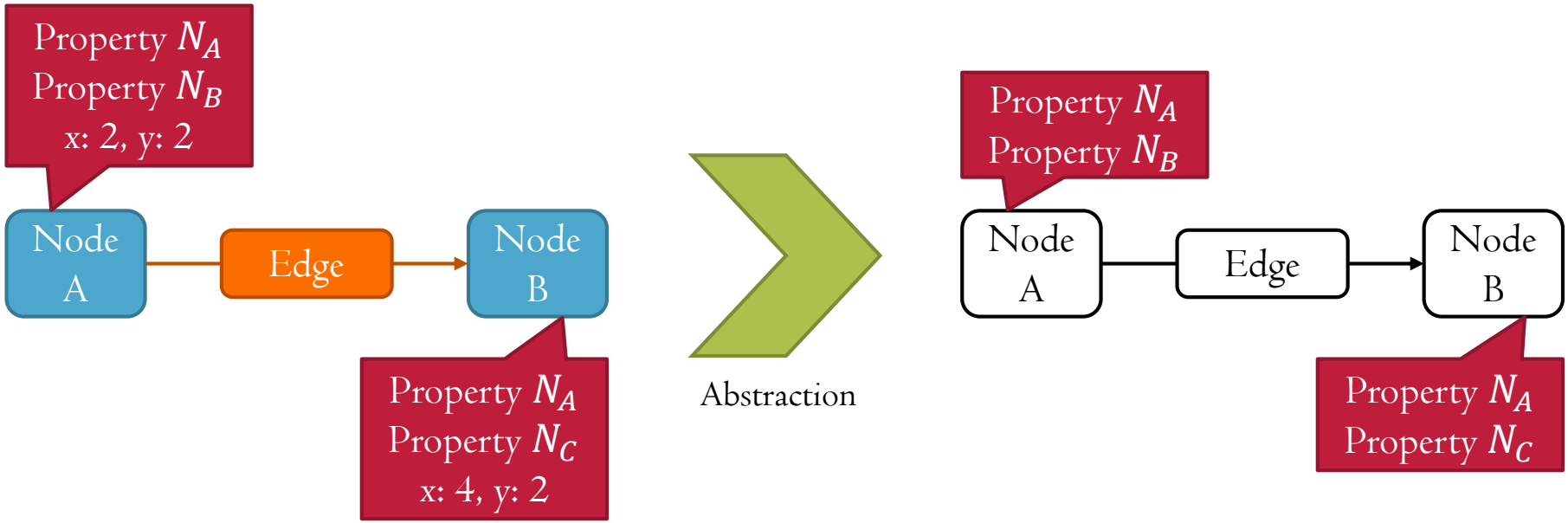
- Nodes
- Edges



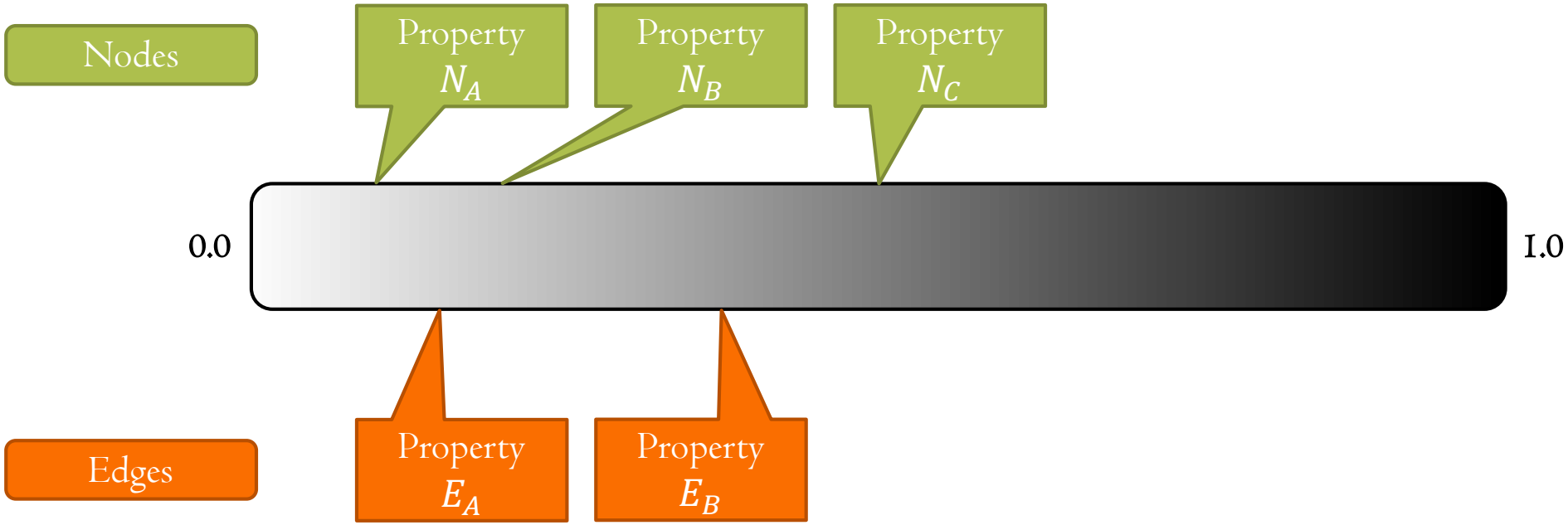
Relevant Properties

- Nodes:
  - Property N_A
  - Property N_B
  - Property N_C
- Edges:
  - Property E_A
  - Property E_B

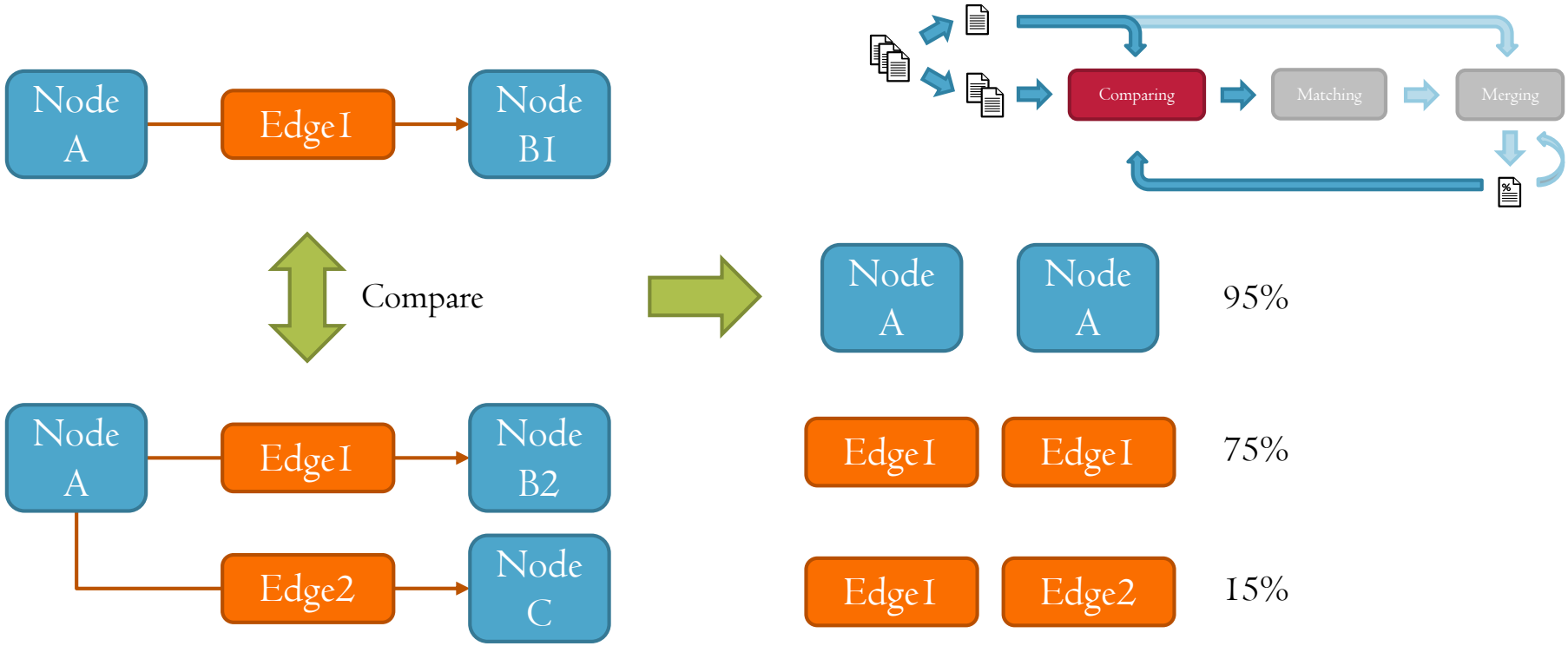
Build a Meta-Model



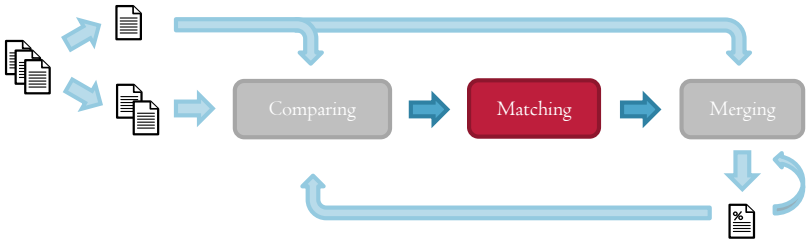
Define a Metric

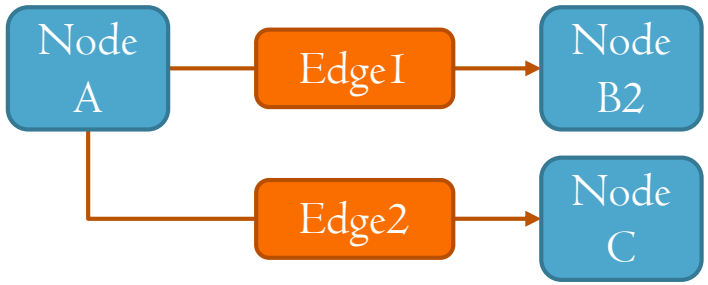


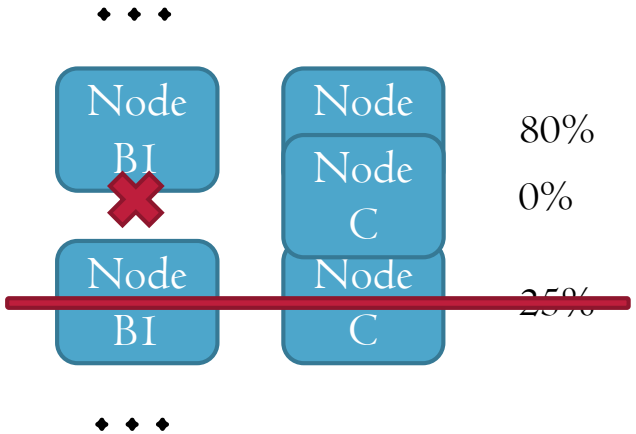
Adapt the Existing Algorithms: Compare



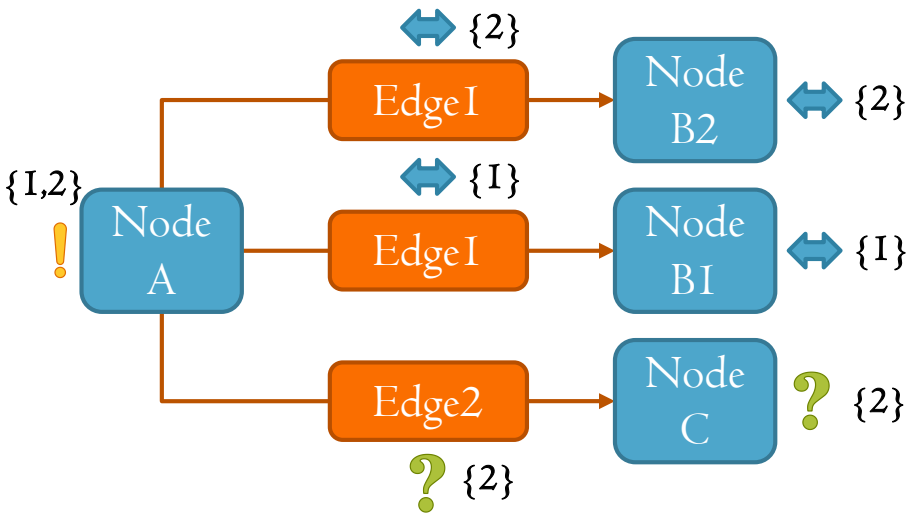
Adapt the Existing Algorithms: Match

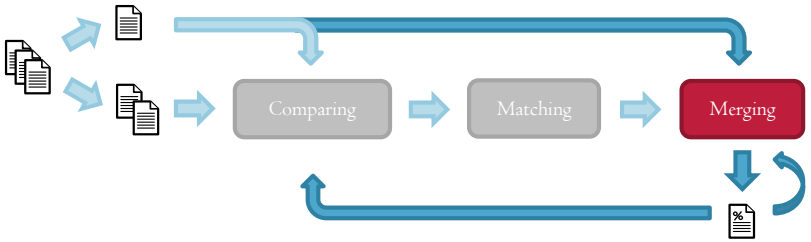


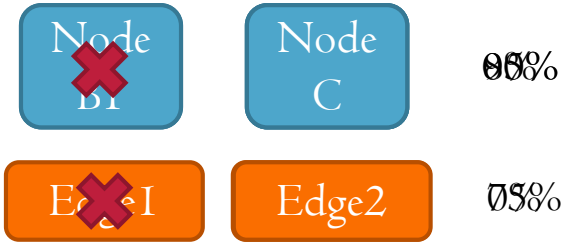




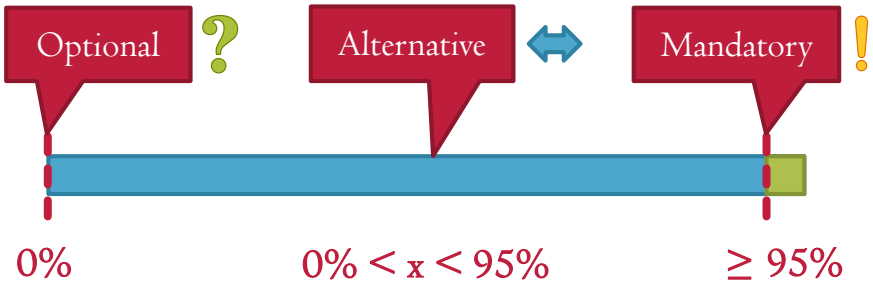
Adapt the Existing Algorithms: Merge



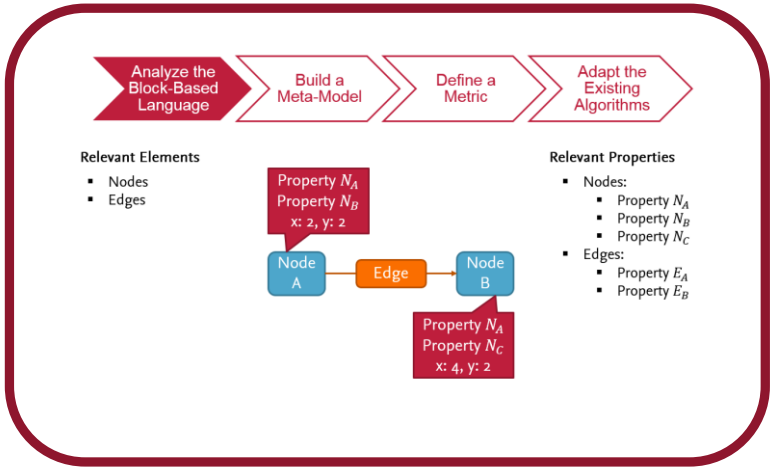


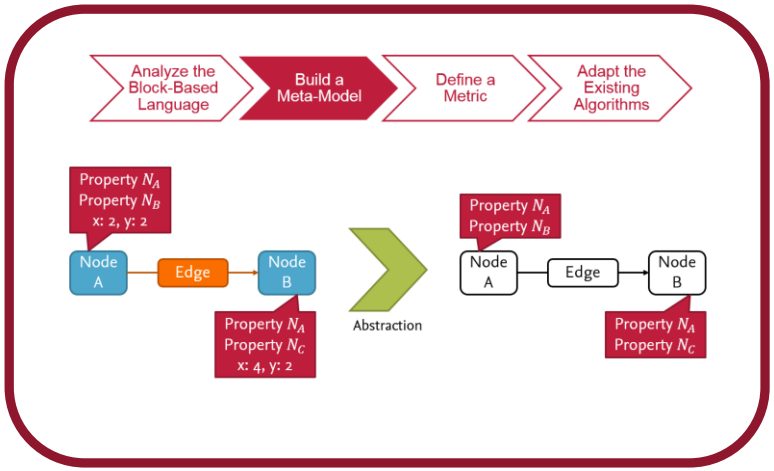


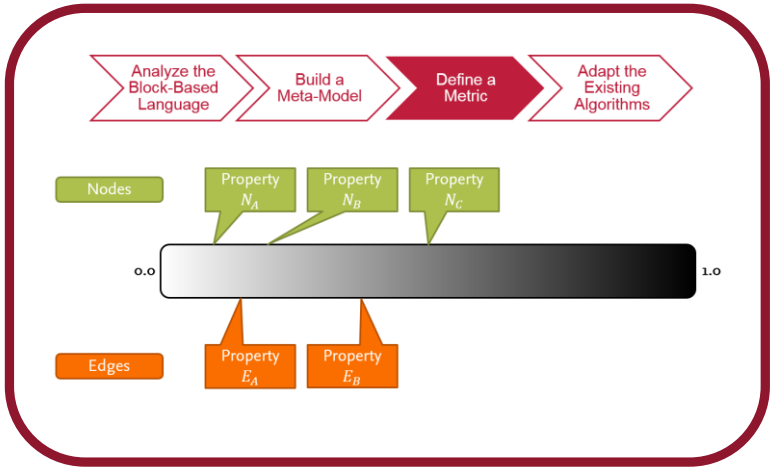
[WHS+13]

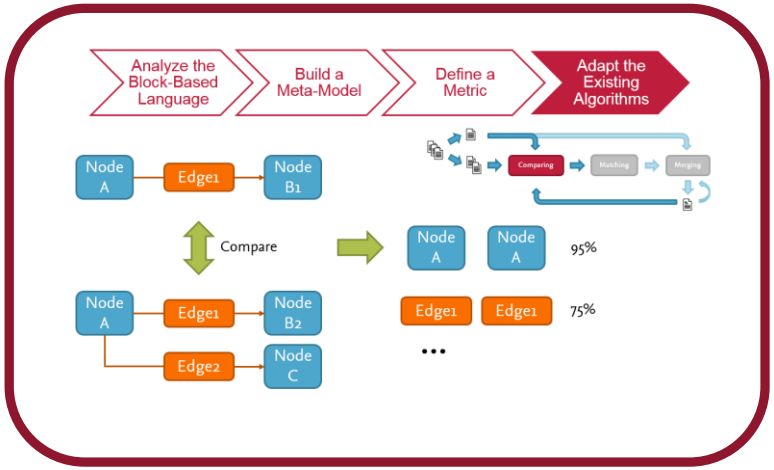


Conclusion


Analyze the Block-Based Language
Build a Meta-Model
Define a Metric
Adapt the Existing Algorithms
Relevant Elements
Nodes
Edges
Relevant Properties
Nodes:
Property N_A
Property N_B
Property N_C
Edges:
Property E_A
Property E_B
Property N_A
Property N_B
x: 2, y: 2
Node A
Edge
Node B
Property N_A
Property N_C
x: 4, y: 2


Analyze the Block-Based Language
Build a Meta-Model
Define a Metric
Adapt the Existing Algorithms
Property N_A
Property N_B
x: 2, y: 2
Node A
Edge
Node B
Property N_A
Property N_C
x: 4, y: 2
Abstraction
Property N_A
Property N_B
Node A
Edge
Node B
Property N_A
Property N_C

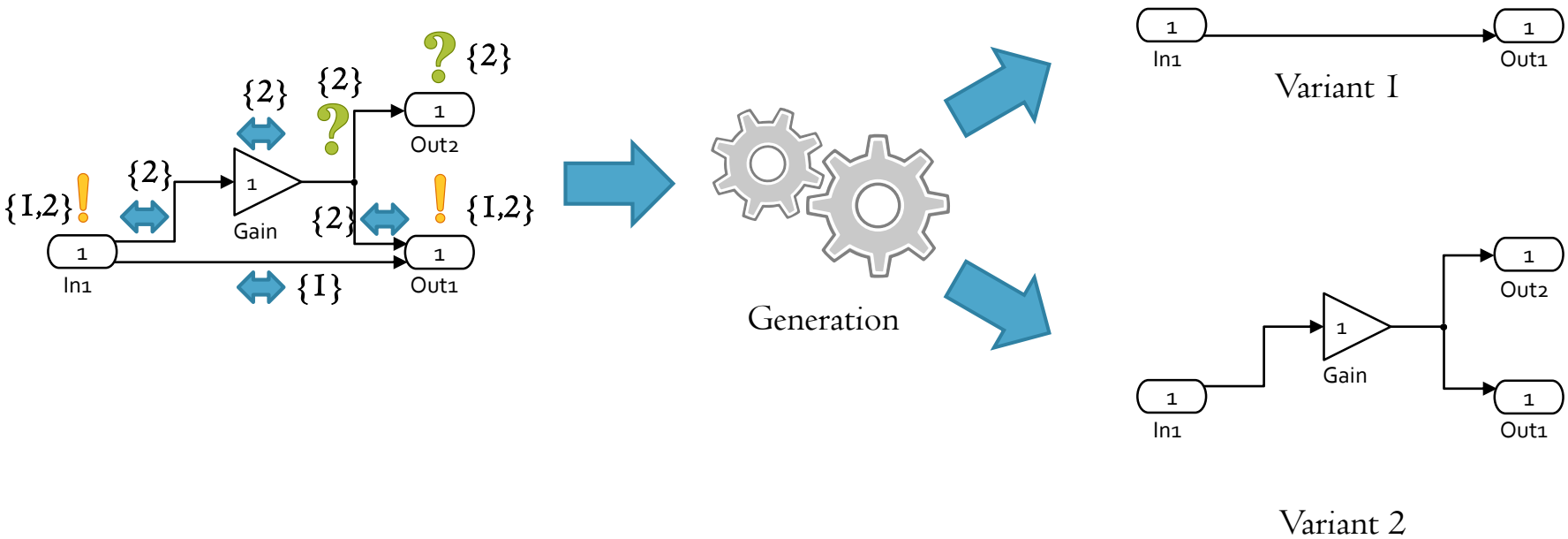

Analyze the Block-Based Language
Build a Meta-Model
Define a Metric
Adapt the Existing Algorithms
Nodes
Property N_A
Property N_B
Property N_C
0.0
1.0
Edges
Property E_A
Property E_B


Analyze the Block-Based Language
Build a Meta-Model
Define a Metric
Adapt the Existing Algorithms
Node A
Edge1
Node B1
Compare
Node A
Edge1
Node B2
Edge2
Node C
Node A
Node A
95%
Edge1
Edge1
75%
...

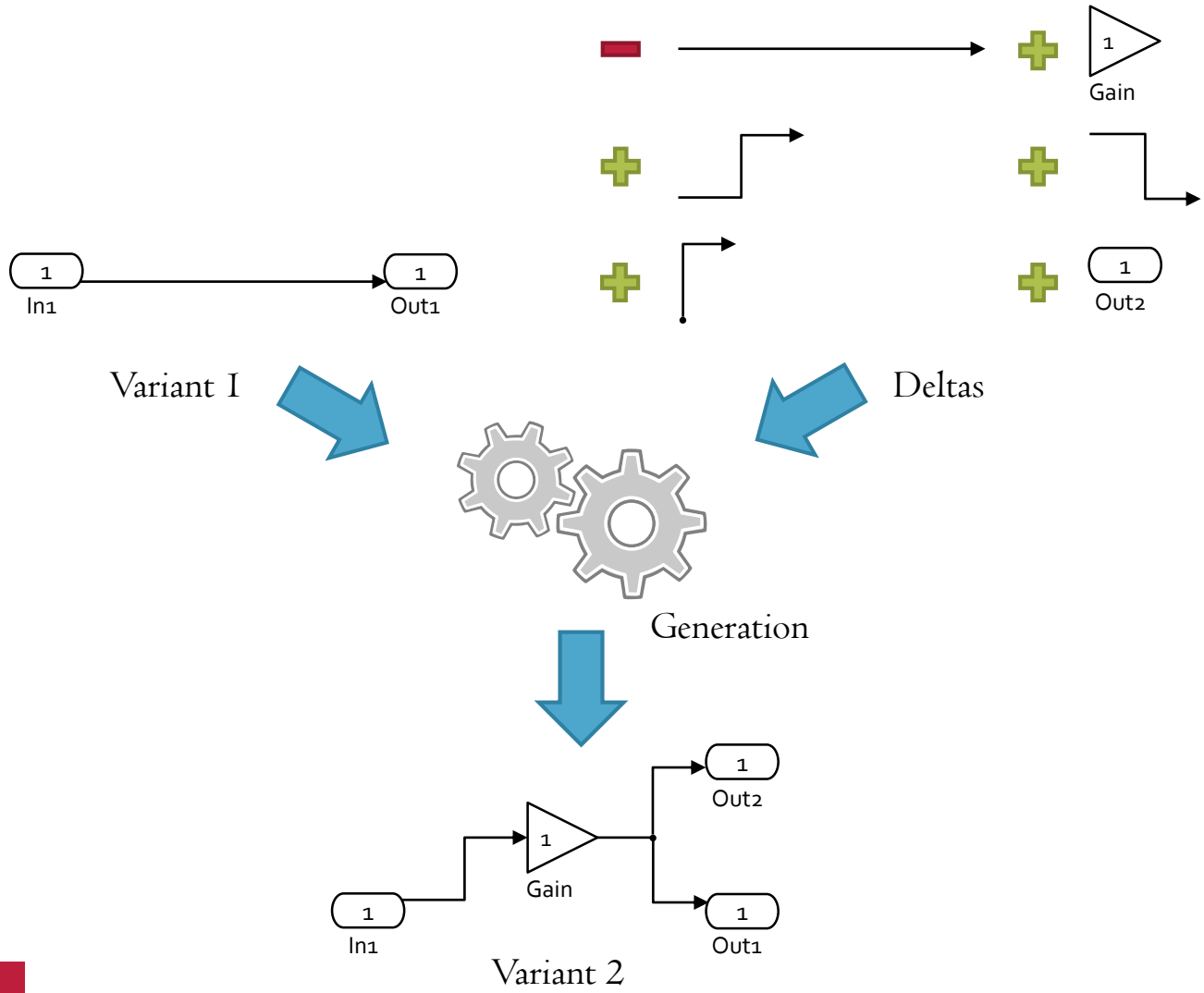
Part 2: Software Product Line Generation

Delta Generation from 150% Models

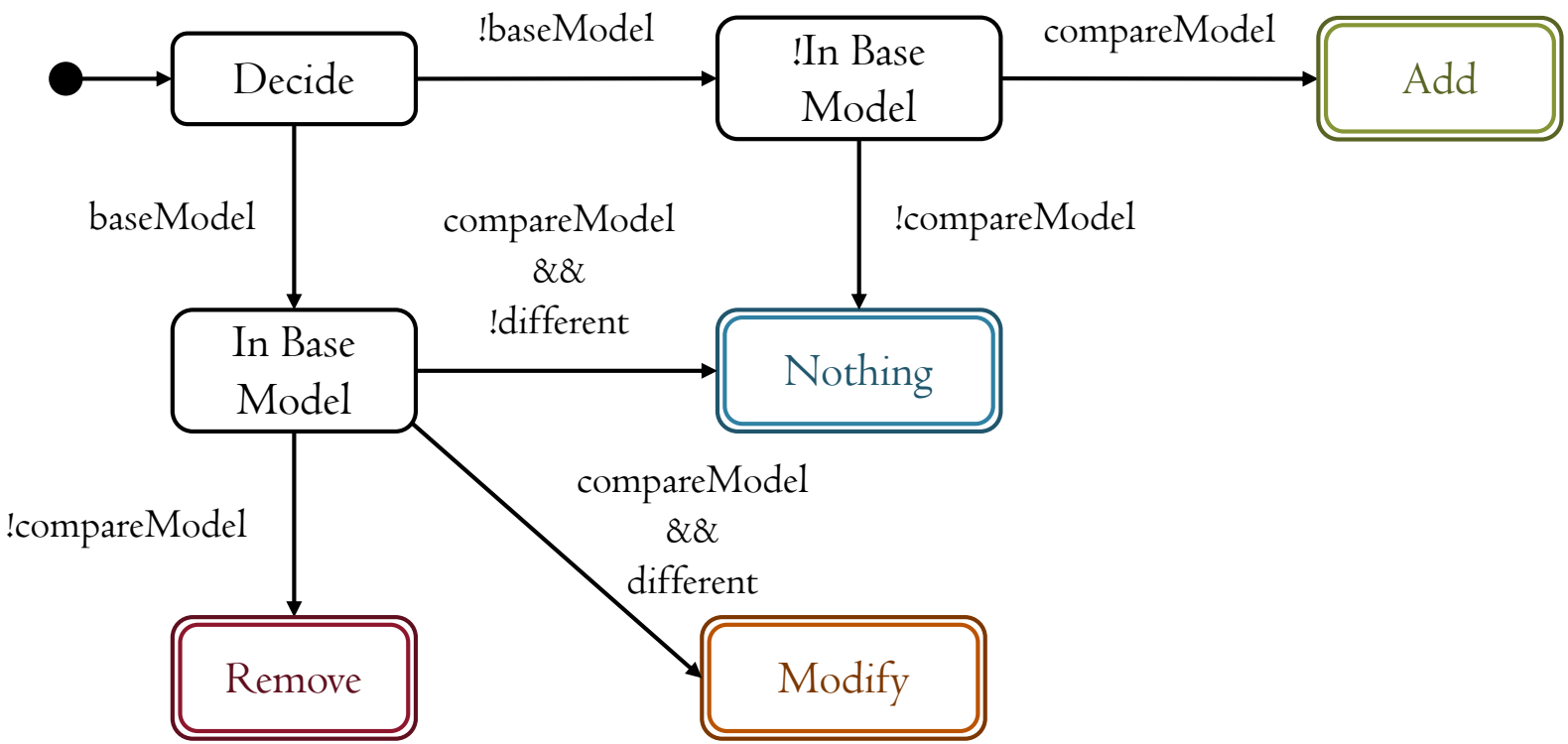
Delta Generation – Motivation



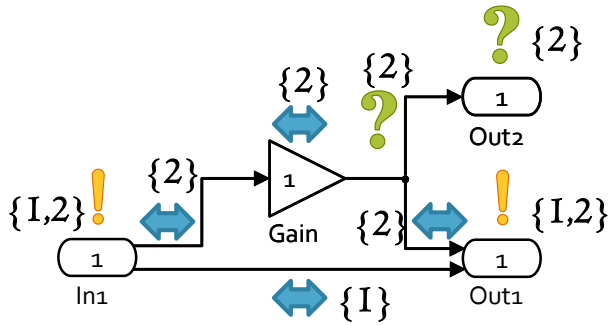
Delta Generation – What are Deltas?



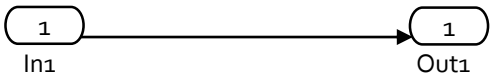
Delta Generation – Concept



Delta Generation – Example

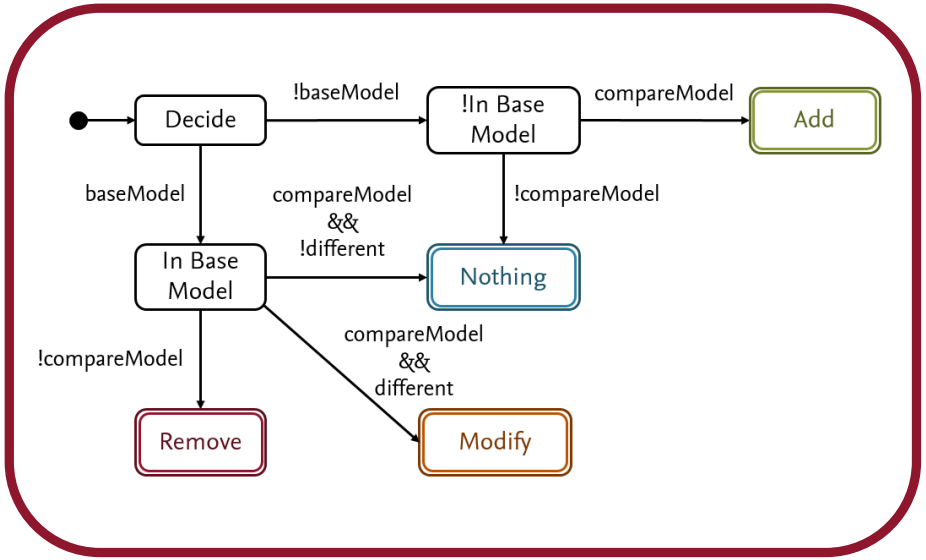


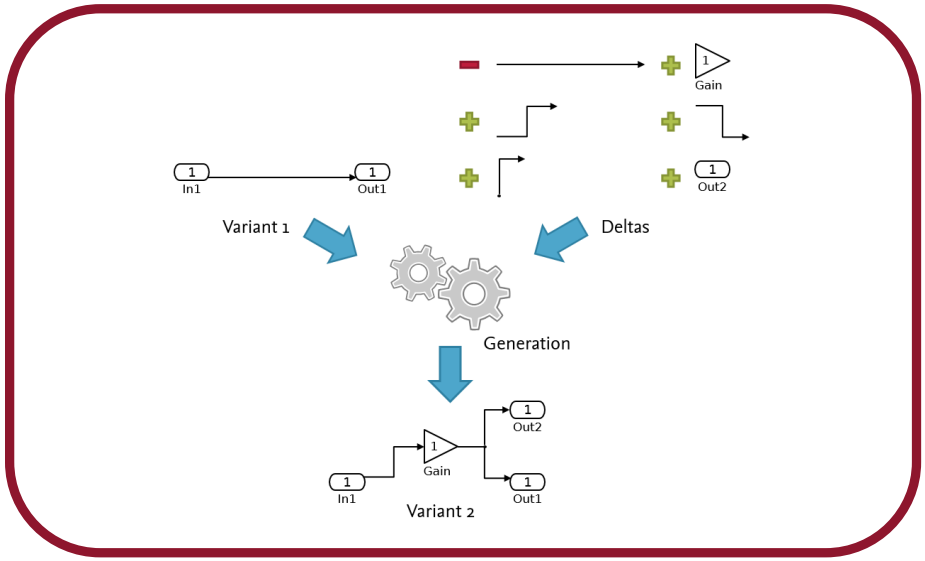
Delta Operations



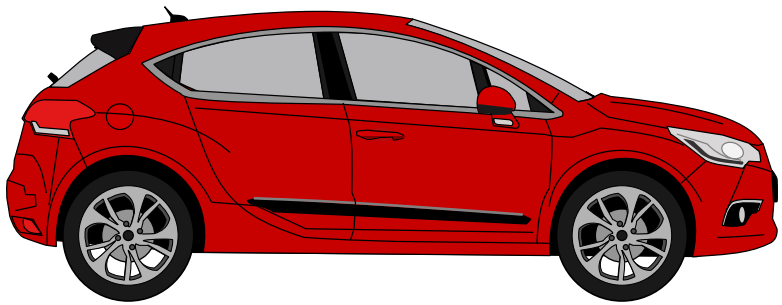
Variant 2

Conclusion

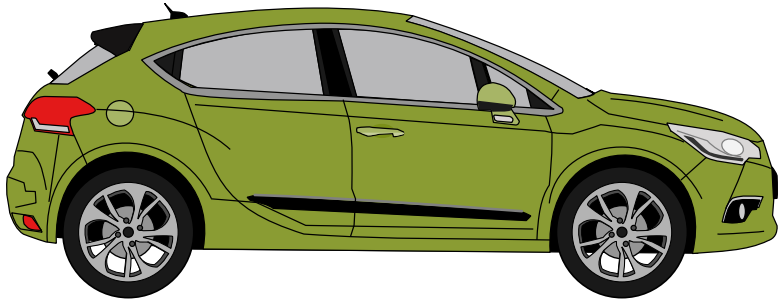

Decide
!baseModel
!In Base Model
compareModel
Add
baseModel
compareModel && !different
!compareModel
In Base Model
Nothing
!compareModel
compareModel && different
Remove
Modify


Gain
In1
Out1
Out2
Variant 1
Deltas
Generation
Variant 2

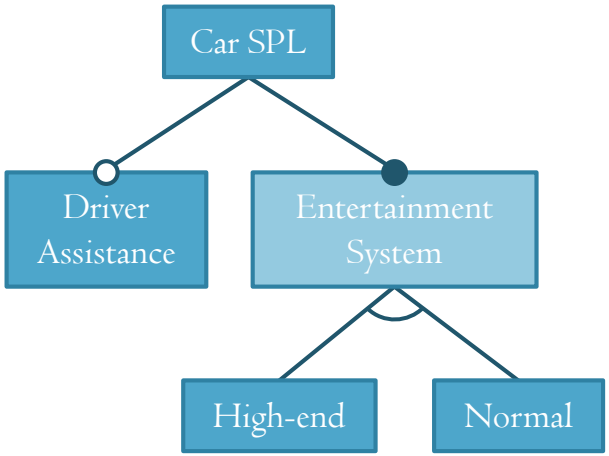
Future Work – Feature Model Generation



Variant 1



Variant 2



Variant 1: {Entertainment System: Normal}

Variant 2: {Entertainment System: High-end, Driver Assistance}

Thank you for your attention!

Are there any questions?

David Wille, M.Sc.
Mail: d.wille@tu-braunschweig.de
Website: https://www.tu-braunschweig.de/isf/team/wille
Phone: +49 (0) 531 391 2288