DFG Deutsche Forschungsgemeinschaft

CONVIDE CRC 1608

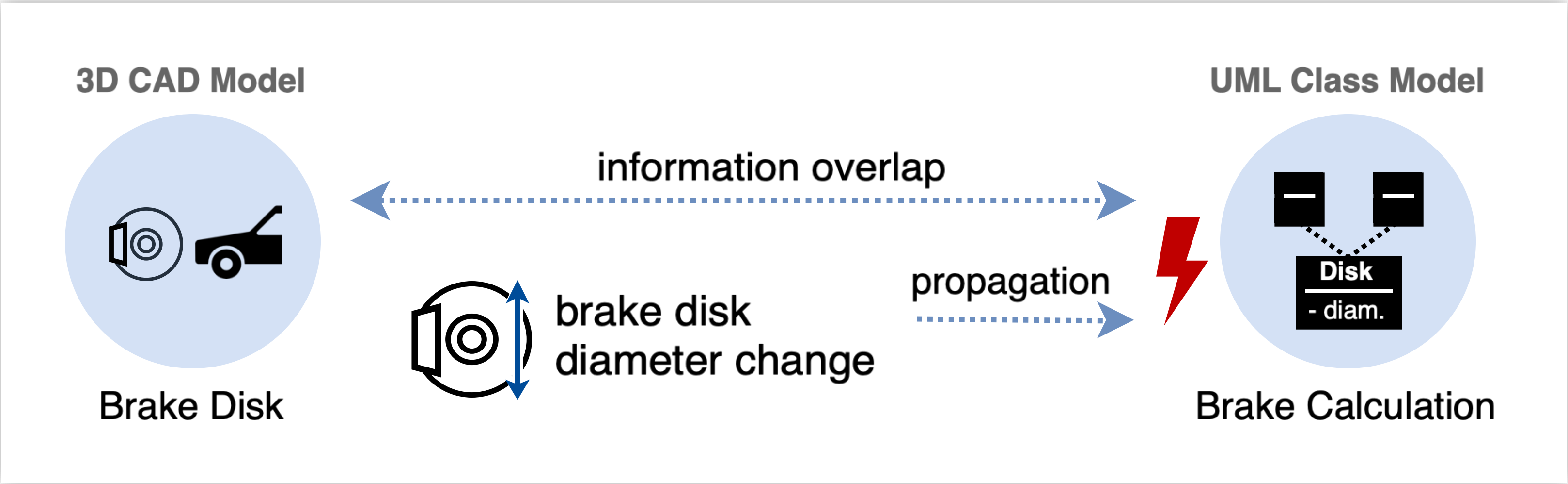
Communicating Changes in Multi-Disciplinary Engineering

Philip Ochs[1], Tobias Pett[1], Lars Gesmann[2], Ina Schaefer[1]

[1]Department of Informatics – Test, Validation and Analysis (TVA)

[2]Department of Mechanical Engineering – Institute of Product Engineering (IPEK)

Model-Based Engineering of Cyber-Physical Systems Motivation



- inconsistencies between models are critical: [1,2]
  - hard to detect (manually & automatically)
  - lead to significant drawbacks & harbour risks for the project’s success

Model-Based Engineering of Cyber-Physical Systems Motivation

- **inconsistencies** between models are critical:
  - hard to detect (manually & automatically)
  - lead to significant drawbacks & harbour risks for the project’s success

- a change to model must be **communicated & propagated** to other engineers across the domains involved
- other engineers initiate subsequent changes to **preserve consistency** of the project

A concise description of a model change is highly domain-specific.
⇒ Hard to understand by engineers outside the outgoing domain.

RQ: How to describe highly domain-specific model changes
in multi-disciplinary engineering?

Languages to Describe Changes – Abstracted

Formal Languages

- concise, unmistakably, mathematically
- processable by computers
- enable automated processes (analyses, transformations, …)

- not really human-interpretable
- … especially outside the software engineering domain

⇒ not suitable to describe & communicate changes in cyber-physical engineering

Informal Languages

- intuitively understandable/ interpretable across all domains and engineers
- naturally enriched by semantics (wording)

- subjective, often expert-based
- not concise, lack standardisation
- to be set up manually, lack basics for automated, computation-based processing

⇒ not suitable to describe & communicate changes in cyber-physical engineering

Idea: Combining the Best of Both Worlds

Formal Languages

- concise, unmistakably, mathematically
- processable by computers
- enable automated processes (analyses, transformations, …)

- use it to specify model-specific changes
  ⇒ formal notion of change

Informal Languages

- intuitively understandable/ interpretable across all domains and engineers
- naturally enriched by semantics (wording)

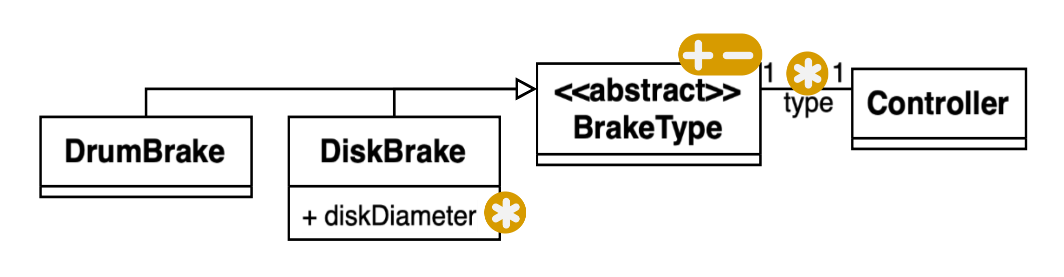
- use it to describe a change model-independently while keeping semantics of a change

Formal Language of Change

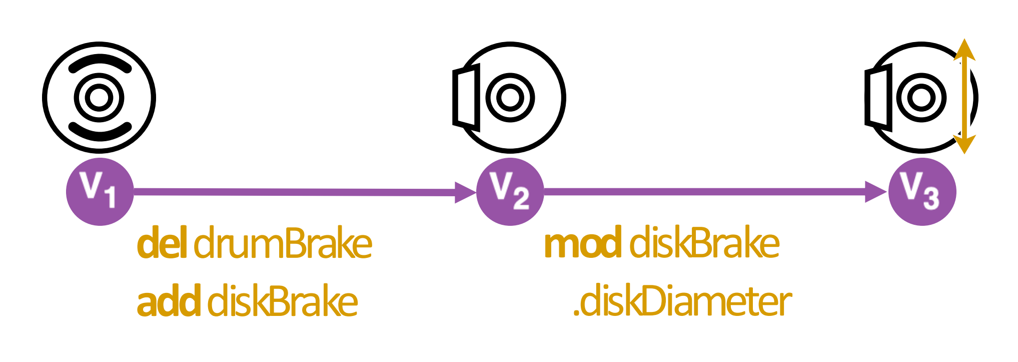
- Concrete Representation: **Delta Modelling** [3]
  - approach to derive product variants in a software product line
  - to a core product variant, apply deltas to get different product variants

	P_0 / Core	P_1	P_2	…
Δ_1		x	x	
Δ_2			x	

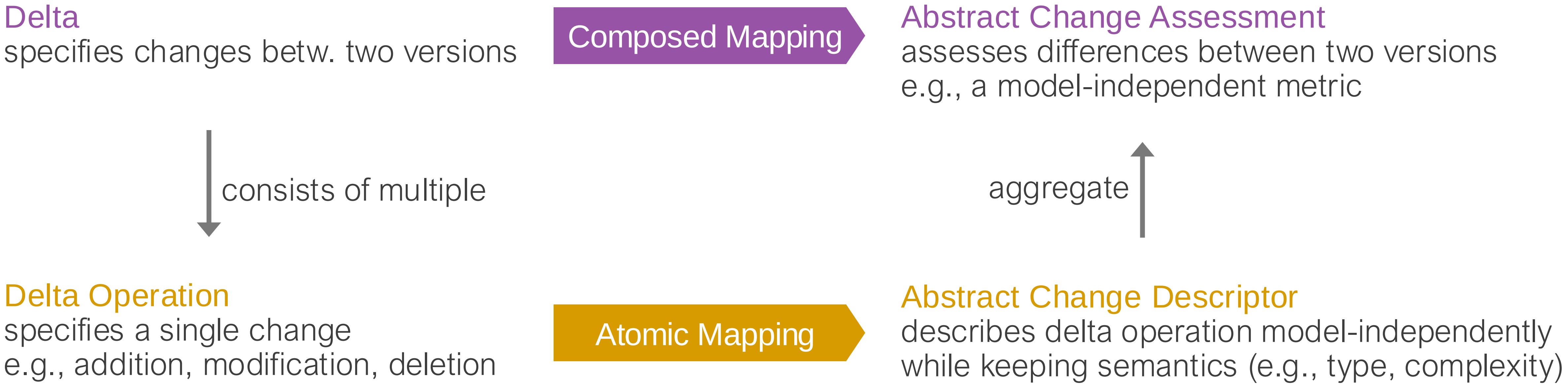
- first adaption: use approach for arbitrary models
  - **delta dialect** defines all possible changes in a model
  - delta dialect itself based on meta model of a model



- second adaption: use approach for variability in time
  - deltas specify changes between two versions
    ⇒ one delta is a set of operations
  - delta operations specify single changes
    e.g., additions, modifications, deletions



Translating Deltas & Delta Operations



Delta Modelling

Delta
specifies changes betw. two versions

consists of multiple

Delta Operation
specifies a single change
e.g., addition, modification, deletion

Composed Mapping

Atomic Mapping

Informal Language

Abstract Change Assessment
assesses differences between two versions
e.g., a model-independent metric

aggregate

Abstract Change Descriptor
describes delta operation model-independently
while keeping semantics (e.g., type, complexity)

Informal Language of Change

- Concrete Representation: Description Model of System Generation Engineering (SGE) [4,5]
  - from the domain of mechanical engineering
  - idea: tracing shares of reused and newly developed parts of a system
  - approach: classify changes into *variation types*

- variation types:
  - hold semantic information of a change
  - important here: variation type c een:
    1. attribute variation (AV)
    2. principle variation (PV)

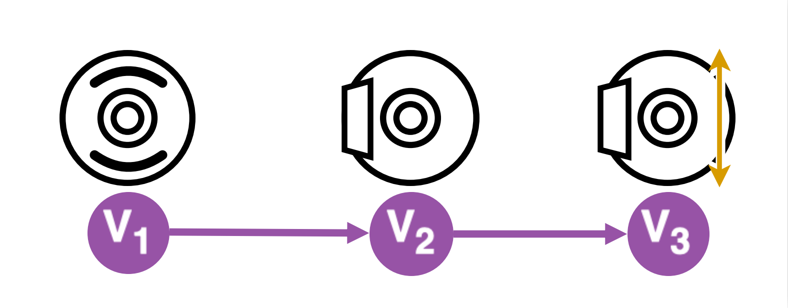
Change Descriptor

low(er) risk ”

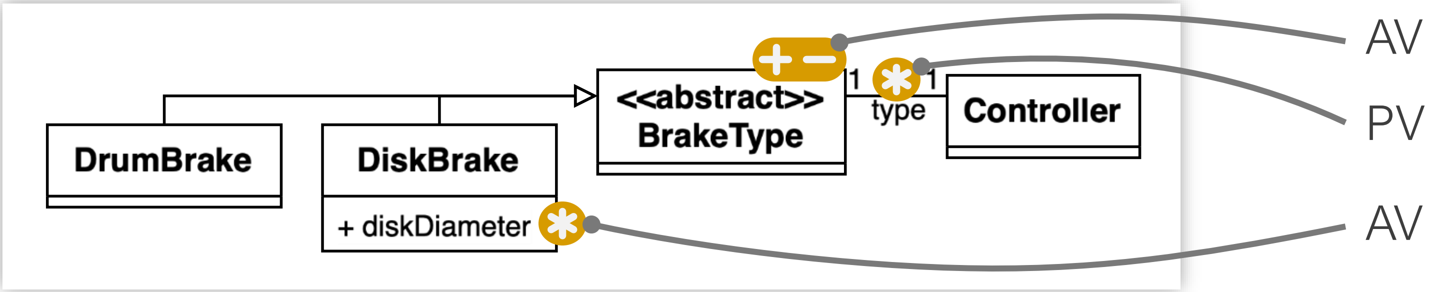
high(er) risk ”

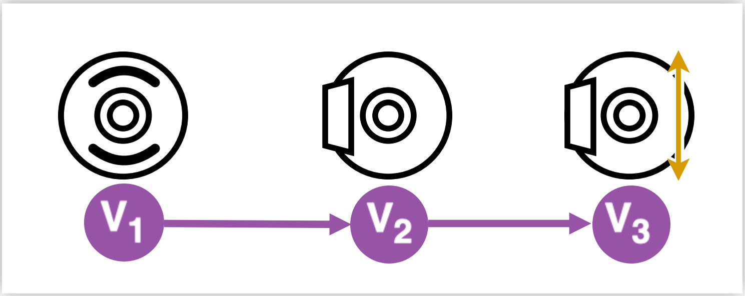
- analysis over sets of variation types:
  - share of reused parts
  - share of newly developed parts

Change Assessment



Full Example





Delta Modelling

- $V_1 \rightarrow V_2$
  add diskBrake
  mod controller.type
  rem drumBrake

- $V_2 \rightarrow V_3$
  mod diskBrake.diskDiameter

Description Model of SGE

- “ $^{2}/_{3}$ changed in V_2, from which $^{1}/_{3}$ in functioning principle “
  AV
  PV
  AV

- “ $^{1}/_{3}$ changed in V_3, from which 0% in functioning principle “
  AV

change of high risk ”

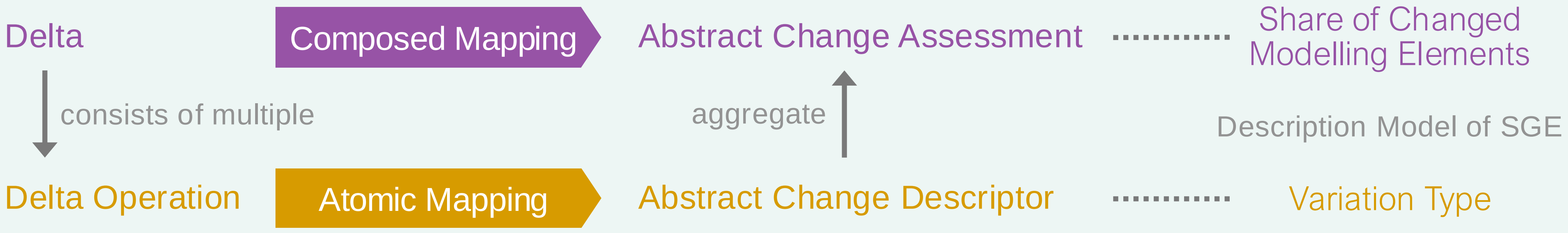
change of low risk ”

Summary & Open Questions

Work in Progress

RQ: How to describe highly domain-specific model changes in multi-disciplinary engineering?

?



Delta

Composed Mapping

Abstract Change Assessment

Share of Changed Modelling Elements

consists of multiple

aggregate

Description Model of SGE

Delta Operation

Atomic Mapping

Abstract Change Descriptor

Variation Type

Q1 [Motivation]: Which aspects of interdiscipl. (change) communication do You think are important to look at?

Q2 [Related Work]: Which approaches do You know for describing changes (formally and informally)?

Q3 [Concept]: How to extend the concept to variable CPS, i.e., variability in time *and* space?

Bibliography

[1] Technical Operations International Council on Systems Engineering (INCOSE). ‘INCOSE Systems Engineering Vision 2020’. INCOSE-TP-2004-004-02, 2007.

[2] Spanoudakis, George, and Andrea Zisman. ‘Inconsistency Management in Software Engineering: Survey and Open Research Issues’, 329–80, 2001. https://doi.org/10.1142/9789812389718_0015.

[3] Schaefer, Ina. ‘Variability Modelling for Model-Driven Development of Software Product Lines.’ VaMoS 10 (2010): 85–92.

[4] Albers, Albert, Nikola Bursac, and Eike Wintergerst. ‘Product Generation Development–Importance and Challenges from a Design Research Perspective’. New Developments in Mechanics and Mechanical Engineering 13 (2015): 16–21.

[5] Albers, Albert, and Simon Rapp. ‘Model of SGE: System Generation Engineering as Basis for Structured Planning and Management of Development’. In Design Methodology for Future Products: Data Driven, Agile and Flexible, edited by Dieter Krause and Emil Heyden, 27–46. Cham: Springer International Publishing, 2022. https://doi.org/10.1007/978-3-030-78368-6_2.

24.03.2025

Philip Ochs, Tobias Pett, Lars Gesmann, Ina Schaefer

Communicating Changes in Multi-Disciplinary Engineering

This Project is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1608 – 501798263