GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft

CONVIDE CRC 1608

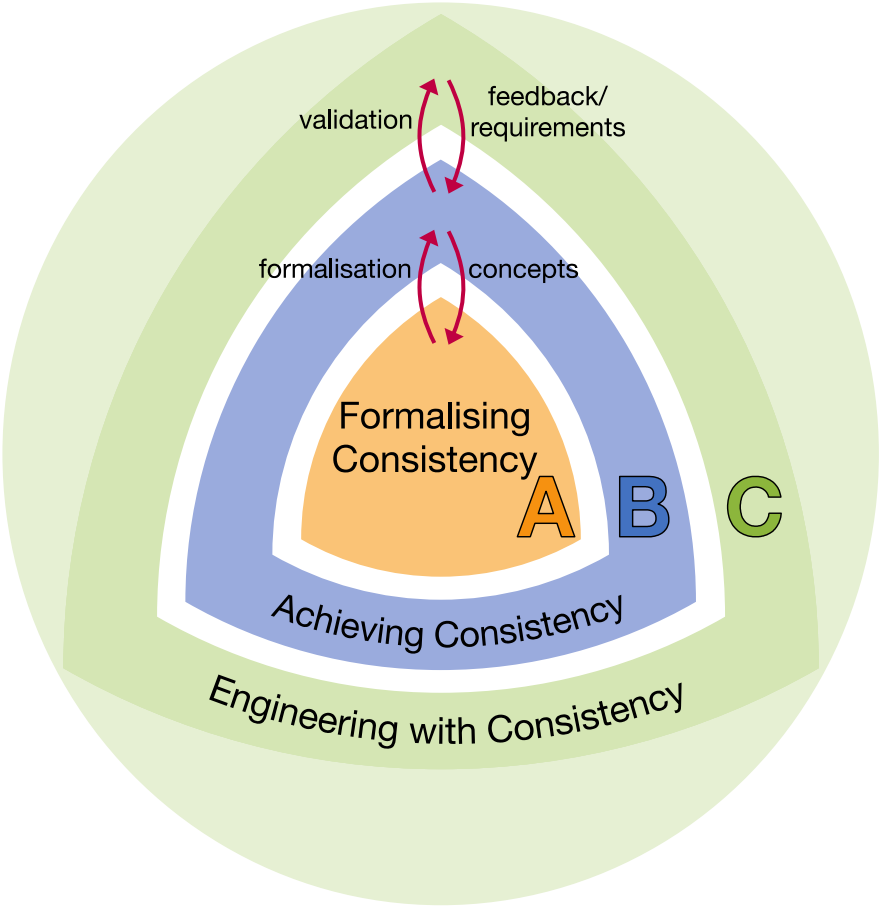
CONVIDE
RESEARCH AREA A

Prof. Dr. Ralf Reussner • Overview • 12.12.2024

ROLE OF RESEARCH AREA A WITHIN CONVIDE

Formalising concepts of consistency in CPS development

- Notions of consistency and their properties:
  - Complexity
  - Certainty
- Special demands of CPS design
  - Hybrid/continuous models
  - Data-defined models



PROJECT OVERVIEW

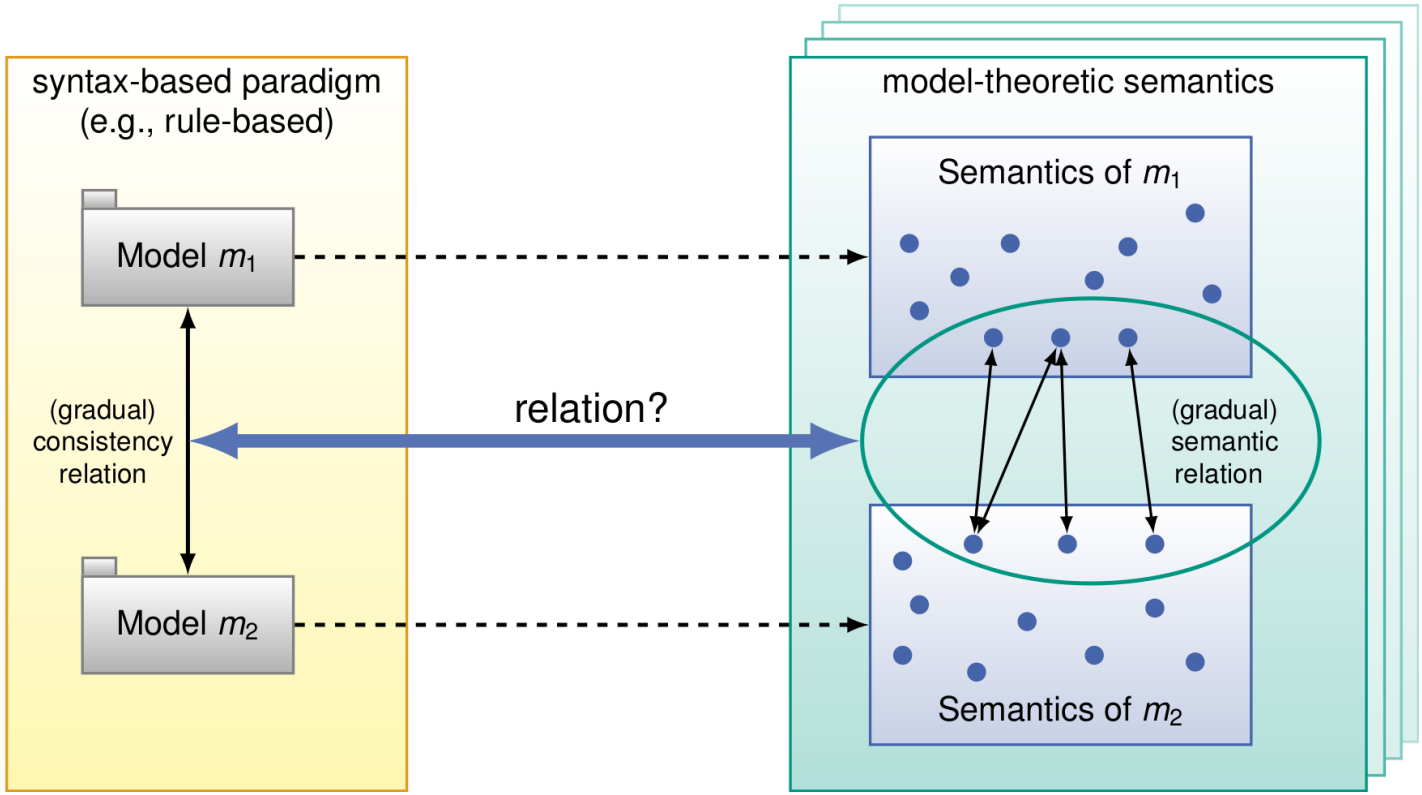
CONVIDE CRC 1608

B01	Cross-Organizational Design of View Types and V-SUM Meta-Models	Atkinson **Pretschner**	
B02	Concurrent Editing and Transactionality	Acosta **Reussner**	
B03	Recovery from Temporary Inconsistency	Koziolek **Ulbrich**	
B04	Maintaining Consistency between Variants and Versions	**Aßmann** Burger Schaefer	

Overview on Projects

A01	Formalising and Relating Different Notions of Consistency	Aßmann **Beckert** Reussner	
A02	Complexity of Consistency	**Atkinson** Burger Ulbrich	
A03	Consistency Under Uncertainty	**Acosta** Koziolek	
A04	Consistency of Hybrid / Continuous Models	Althoff **Platzer** Pretschner	
A05	Consistency of Data-Defined Models	**Althoff** Platzer	

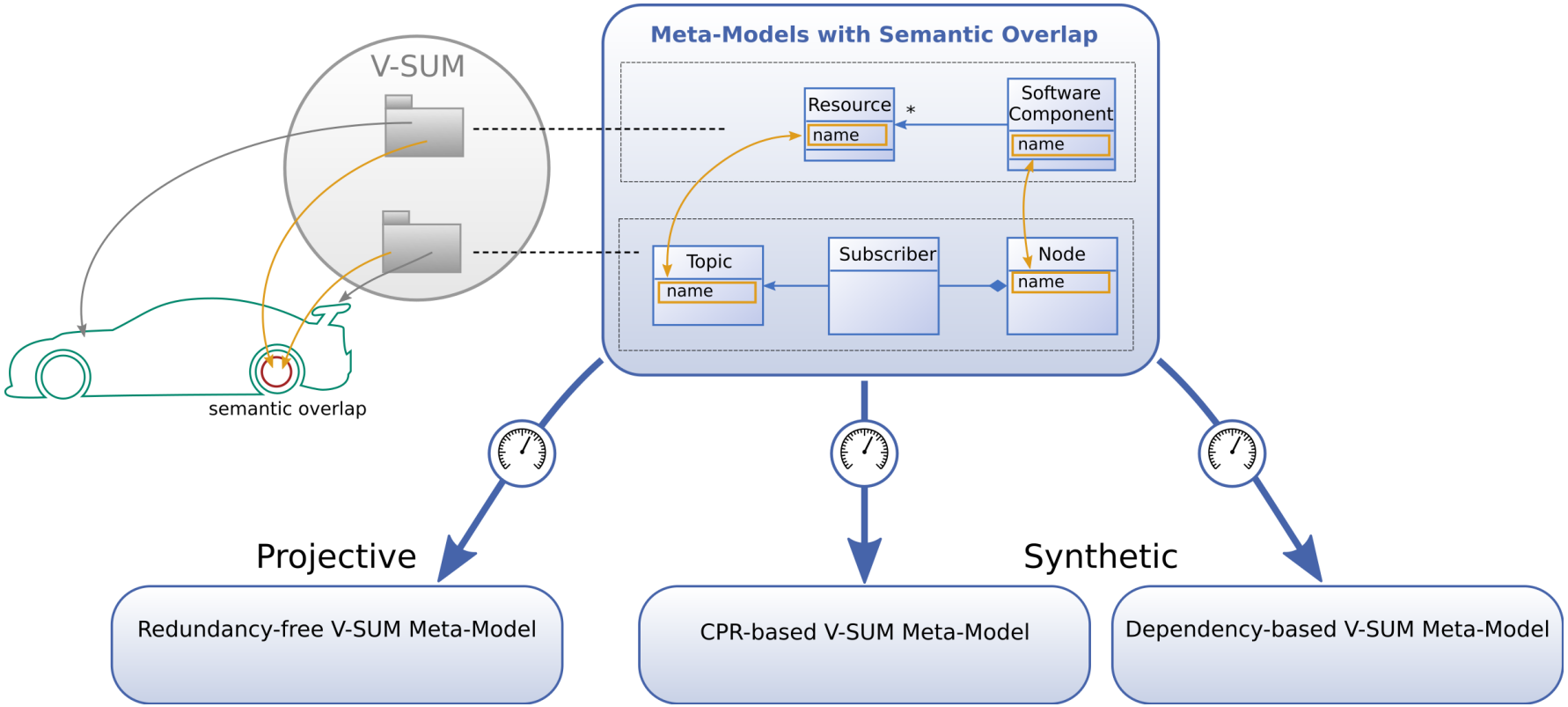
A01: Formalising and Relating Different Notions of Consistency



Exemplary Research Question

What are properties and relations between different notions of consistency?

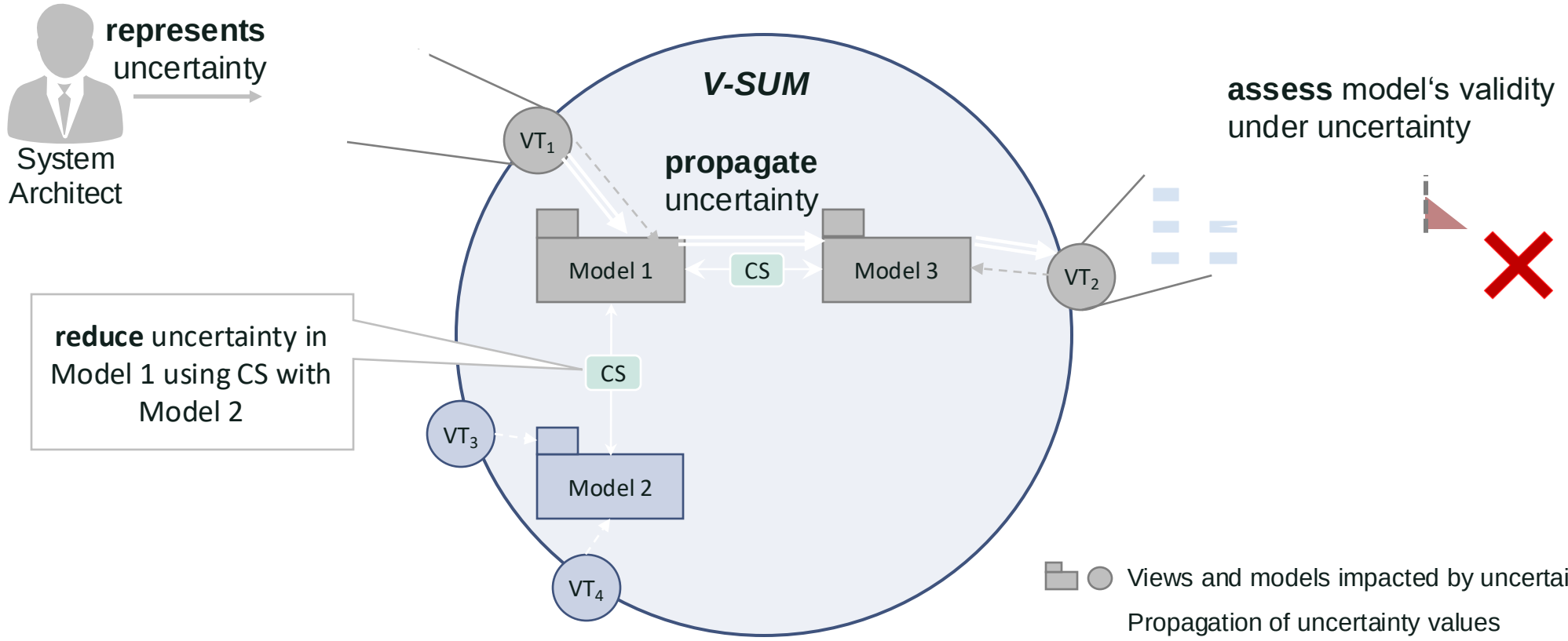
A02: Complexity of Consistency



Exemplary Research Question

How can the complexity and impact of the different semantic overlap resolution approaches be measured to support the design of a V-SUM meta-model?

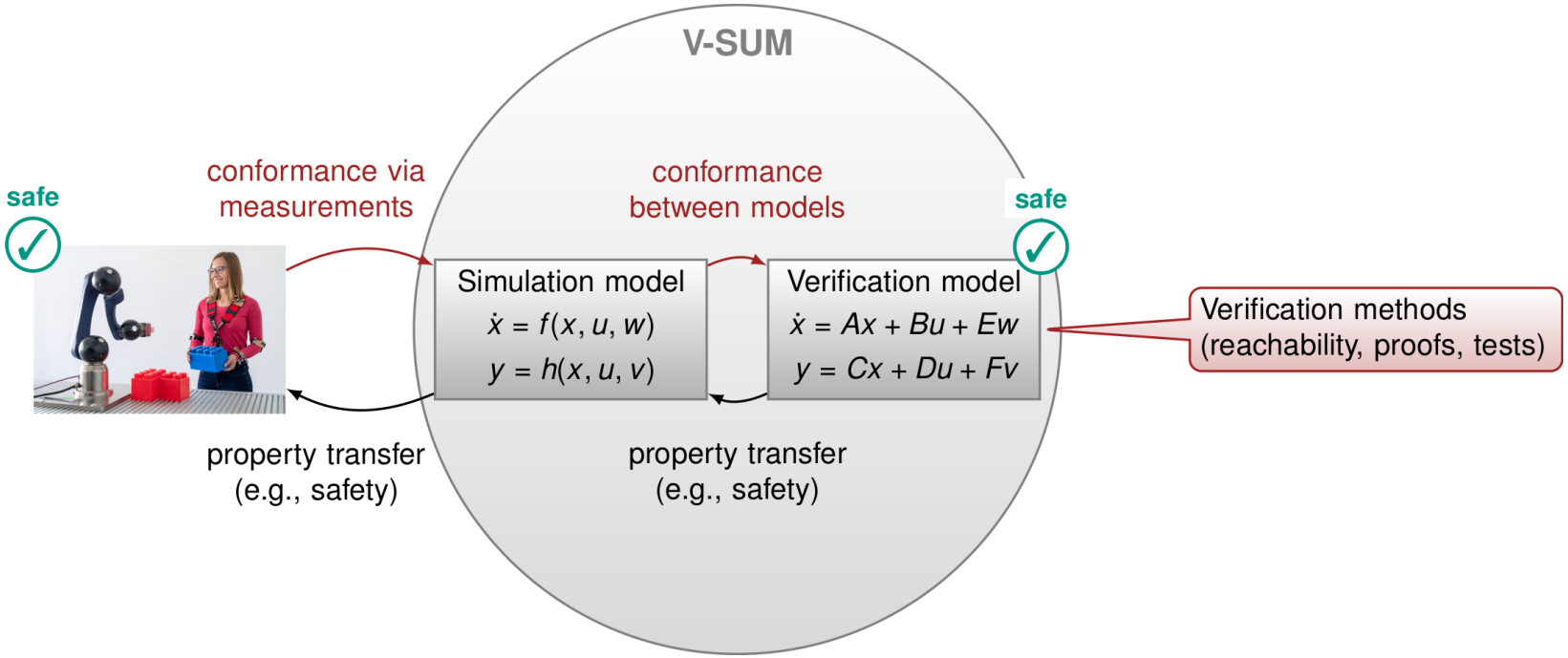
A03: Consistency Under Uncertainty



Exemplary Research Questions

How can uncertainty in CPS design be dealt with? What does it mean for consistency?
How could consistency relations be used to lower uncertainty?

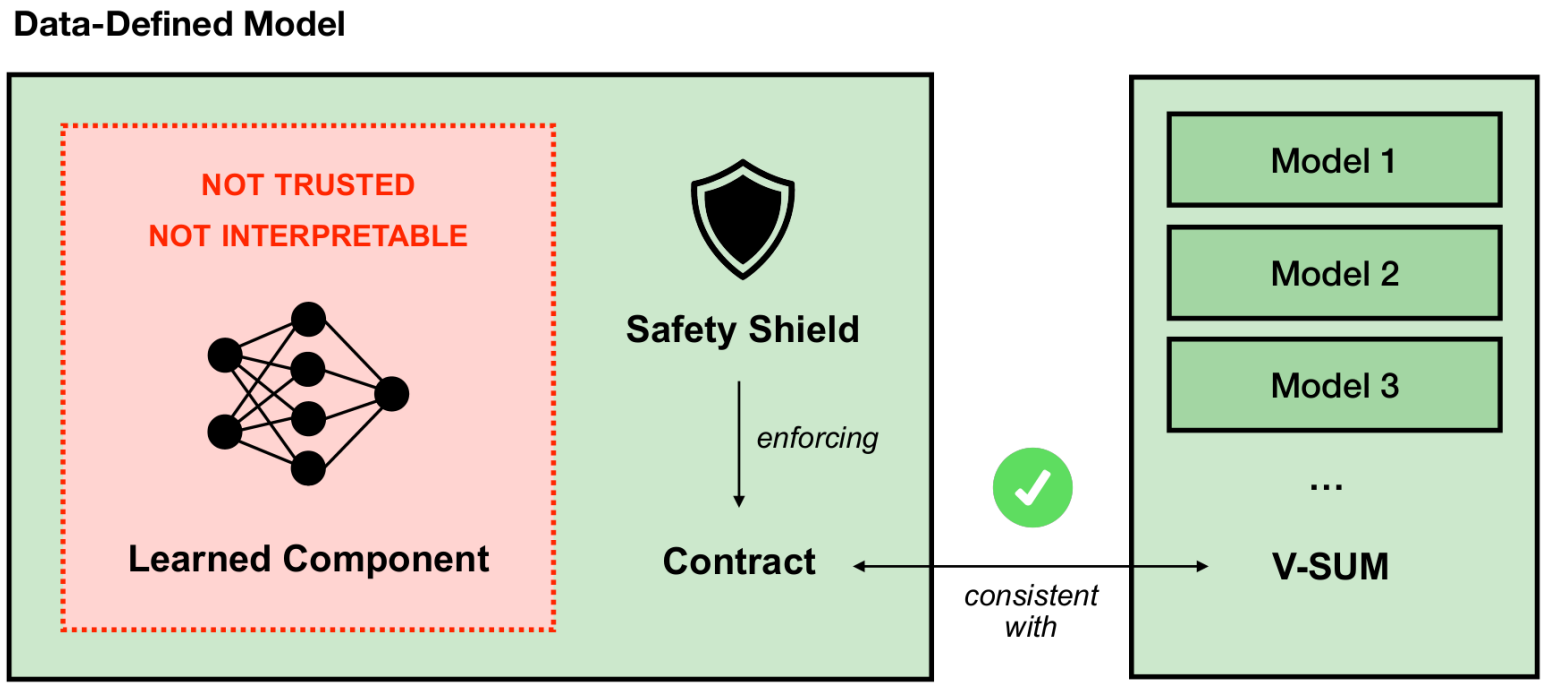
A04: Consistency of Hybrid / Continuous Models



Exemplary Research Question

How to specify consistency for hybrid models?

A05: Consistency of Data-Defined Models



Exemplary Research Question

What does consistency mean for data-defined models?

Highlight Talks Today

- A01: Romain Pascal: Multidimensional Consistency, Looking into Semantics
- A02: Colin Atkinson: Towards Deep Reactions in Vitruvius
- A04: André Platzer: Differential Refinement Logic for Hybrid Systems Consistency

Karlsruhe Institute of Technology

CONVIDE
CRC 1608

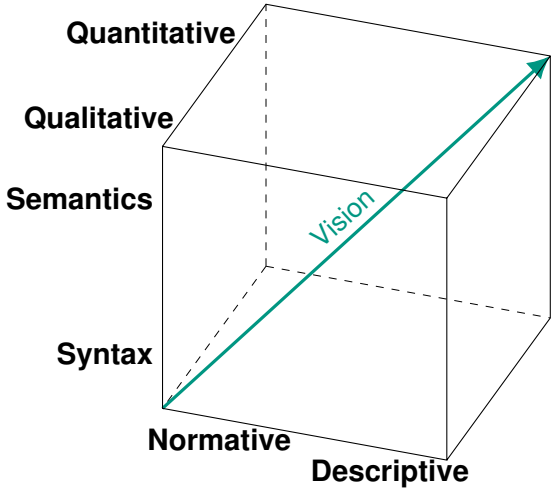
A01 – Multidimensional Consistency, Looking into Semantics

Winter Colloquium

U. Aßmann, B. Beckert, K. Feichtinger, K. Kegel, R. Pascual, R. Reussner | Decembre 12, 2024

Multidimensional Consistency[1]

- **Binary vs. N-ary**
  Reason about multiple models
- **Normative vs. Descriptive**
  Reason about correctness
- **Qualitative vs. Quantitative**
  Reason about consistency-increasing methods
- **Certainty vs. Uncertainty**
  Reason about the physical part of the system
- **Syntax vs. Semantics**
  Reason about quality



[1] Feichtinger et al. 2024.

Semantics

Semantics with Java programs as models

- **trace semantics**
- **pre** and **post** conditions
- **result** of **tests**
- **termination** property
- **number of methods** or **attributes** of a class

Semantics

Semantics with Java programs as models

- **trace semantics**
- **pre** and **post** conditions
- **result** of **tests**
- **termination** property
- **number of methods** or **attributes** of a class

Abstract semantics

$$[\![\cdot]\!] : M \to S$$

M meta-model and S semantic space

It is purpose-dependent

Main findings[2]

- **1.** Imposing conditions on the semantic spaces allows for a notion of semantical V-SUM.

[2] Pascual et al. 2024.

Main findings[2]

- **1.** Imposing conditions on the semantic spaces allows for a notion of semantical V-SUM.
- **2.** There exist semantics called **natural semantics** that capture exactly the information necessary to assess the consistency of models.

[2] Pascual et al. 2024.

Example

Suppose that $(m_i \in M_i)_{i \in I}$ describe **components** of a car

The models are **consistent** if the total weight is $\leq$ **1000** kg

The natural semantics are

$$[\![\cdot]\!]_i^{\text{nat}} \colon M_i \to [0, 1000] \cup \{\text{too much}\}$$

References I

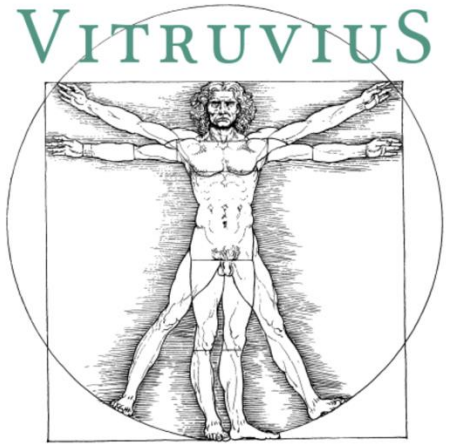
[1] Kevin Feichtinger et al. “Towards Formalizing and Relating Different Notions of Consistency in Cyber-Physical Systems Engineering”. In: **Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems**. MODELS Companion ’24. New York, NY, USA: Association for Computing Machinery, Oct. 31, 2024, pp. 915–919. ISBN: 979-8-4007-0622-6. DOI: 10.1145/3652620.3688565.

[2] Romain Pascual et al. “Formal foundations of consistency in model-driven development”. In: **12th international symposium on leveraging applications of formal methods, verification and validation (ISoLA 2024)**. Lecture notes in computer science. Oct. 2024.

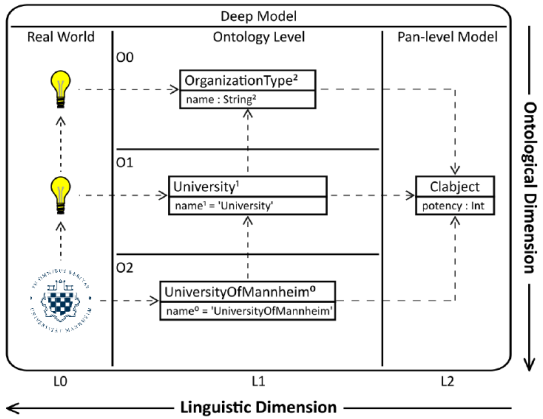
GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft – SFB-1608 – 501798263

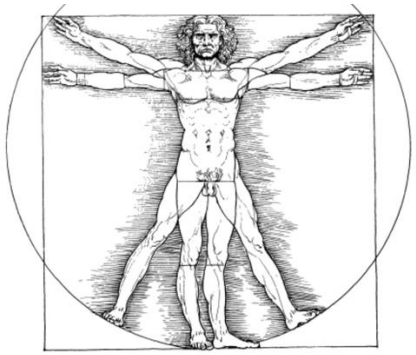
CONVIDE CRC 1608

A2: Colin Atkinson: Towards Deep Reactions in Vitruvius

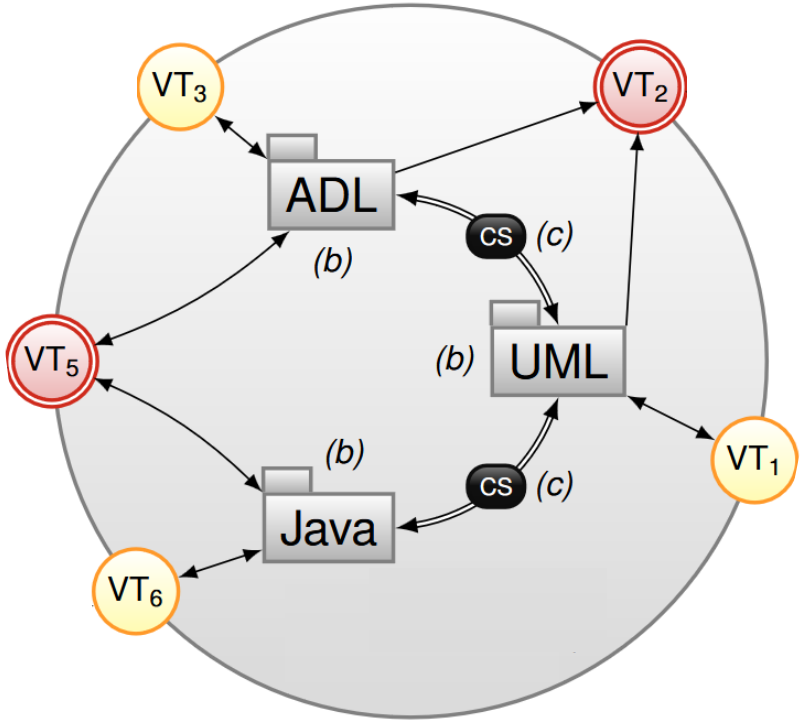


Multi-level (Deep) Modeling





Multi-level (Deep) Vitruvius



Vitruvius VSUM

```
reactions: umlToJavaClass
in reaction to changes in uml
execute actions in java

reaction CreatedUmlClass {
    after element uml::Class
        created and inserted as root
    call {
        val umlClass = newValue
        createJavaClass(umlClass)
    }
}

routine createJavaClass(uml::Class umlClass) {
    match { /* retrieve_elements */ }
    create { /* create_elements */ }
    update { /* update_models */ }
}
```

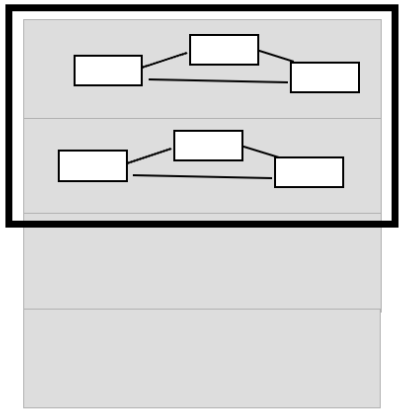
Reactions Declaration

Reactions Definition

Routine Definition

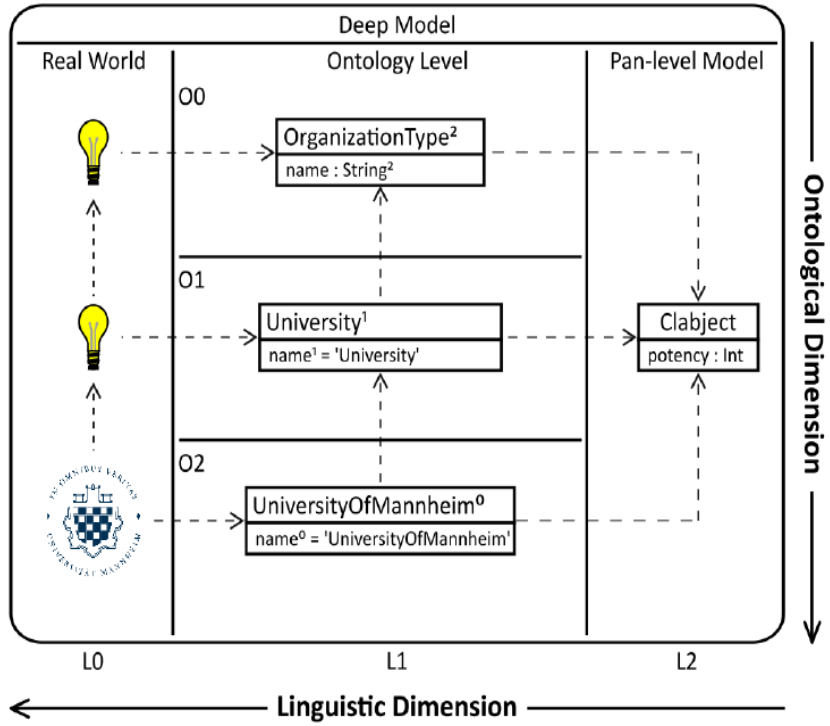
Consistency Specification Rules

The Reactions Language



Define

Covering multiple domain levels with „two-level" modeling technology



Covering multiple domain levels multi-level modeling technology

```
1  import deep "pathtolml_model" as owner
2  import deep "pathtolml_model" as supporter
3
4  reactions: owner2supporter
5  in reaction to changes in owner below level 1
6  execute actions in supporter below level 1
7
8  reaction NewS400Inserted {
9      after direct element owner::S400 inserted in owner at level 2
10
11 reactions: owner2supporter
12 in reaction to changes in owner below level 1
13 execute actions in supporter below level 1
14
15 routine addNewS400(owner::S400 oldS400) {
16     match {
17         val supDeviceLevel = retrieve level 2 in supporter
18     }
19     create {
20 reaction NewS400Inserted {
21     after direct element owner::S400 inserted in owner at level 2
22
23         supDevice.name = oldS400.name
24         addCorrespondenceBetween(oldS400, supDevice)
25         supDeviceLevel.content.add(supDevice)
26     }
27 }
```

- **Import of Deep Models using ‚deep‘**
  - Support for Deep Types

Restriction of changes to certain levels

Making Reactions Level Aware

GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft – SFB-1608 – 501798263

CONVIDE CRC 1608

MULTI@MODELS 24 • September 22–27, 2024, Linz, Austria

TOWARDS DEEP REACTIONS IN MULTI-LEVEL, MULTI-VIEW MODELING

Thomas Weber, Arne Lange, Erik Burger, Lars König, Martin Armbruster

Karlsruher Institut für Technologie

Colin Atkinson, **Monalisha Ojha**, Mohammad Sadeghi

UNIVERSITY OF MANNHEIM

Towards Deep Reactions in Multi-Level, Multi-View Modeling

Thomas Weber
KASTEL
Karlsruhe Institute of Technology
Karlsruhe, Germany
thomas.weber@kit.edu

Monalisha Ojha
Software Engineering Group
University of Mannheim
Mannheim, Germany
monalisha.ojha@uni-mannheim.de

Mohammad Sadeghi
Software Engineering Group
University of Mannheim
Mannheim, Germany
mohammad.sadeghi@uni-mannheim.de

Lars König
KASTEL
Karlsruhe Institute of Technology
Karlsruhe, Germany
lars.koenig@kit.edu

Martin Armbruster
KASTEL
Karlsruhe Institute of Technology
Karlsruhe, Germany
martin.armbruster@kit.edu

Arne Lange
Software Engineering Group
University of Mannheim
Mannheim, Germany
lange@uni-mannheim.de

Erik Burger
KASTEL
Karlsruhe Institute of Technology
Karlsruhe, Germany
burger@kit.edu

Colin Atkinson
Software Engineering Group
University of Mannheim
Mannheim, Germany
atkinson@uni-mannheim.de

ABSTRACT

As the scale, complexity, and scope of software-intensive systems continue to grow, so does the importance of synergistically integrating two important emerging paradigms in software engineering - multi-level modeling and multi-view modeling. While stable tooling for both has been developed by research institutions in recent years, to date no tool has attempted to integrate the two at a fundamental level. In this paper, we describe some first steps we have taken in this direction by integrating the Vitruvius V-SUM-based multi-view environment with the Melanee multi-level modeling environment. In particular, we show how Vitruvius's Reactions language, which allows different models in Vitruvius V-SUMs to be kept consistent, can be extended to support multi-level V-SUMs and views represented in Melanee's dialect of multi-level modeling.

CCS CONCEPTS

• **Software and its engineering** → **Domain specific languages**; *Specialized application languages*; Application specific development environments; • **Information systems** → *Mediators and data integration.*

KEYWORDS

Multi-level modeling, V-SUM, View-based modeling, Vitruvius, Consistency

MODELS Companion '24, September 22–27, 2024, Linz, Austria

ACM ISBN 979-8-4007-0622-6/24/09
https://doi.org/10.1145/3652620.3688208

ACM Reference Format:
Thomas Weber, Monalisha Ojha, Mohammad Sadeghi, Lars König, Martin Armbruster, Arne Lange, Erik Burger, and Colin Atkinson. 2024. Towards Deep Reactions in Multi-Level, Multi-View Modeling. In *ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3652620.3688208

1 INTRODUCTION

As software systems have grown in size and complexity, and are developed as parts of integrated cyber-physical systems, it has become increasingly important to be able to describe and model them using interrelated collections of so-called *views*. View-based modeling environments that keep large numbers of semantically overlapping descriptions of systems consistent over time are therefore receiving growing attention in academia and industry. Of the two basic strategies for achieving inter-view consistency, the so-called projective approach is the most promising at scale, since it reduces the number of pairwise consistency relationships that need to be maintained [9]. However, it requires some kind of central megamodel, or *Single Underlying Model* (SUM) to serve as the source of information and truth from which the views can be projected.

The Vitruvius framework is one such environment that supports the projective approach using a *Virtual* SUM (i.e., V-SUM) rather than a pure, redundancy-free SUM. This obviates the daunting challenge of creating a pure SUM in real-life software engineering projects where it is necessary to work with and integrate, many existing models, based on long-established and utilized metamodels. A Vitruvius V-SUM therefore facilitates the consistent connection of multiple, semantically overlapping models and metamodels by means of *Consistency Preservation Rules* (CPRs) written in a specially designed Reactions language.

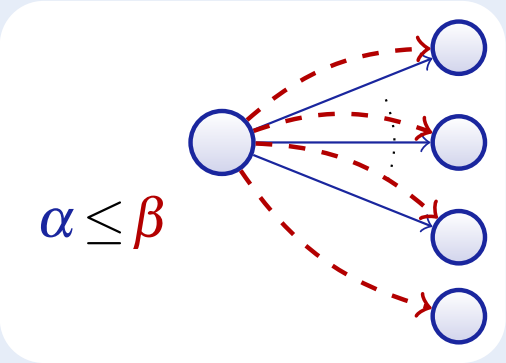
Differential Refinement Logic for Hybrid Systems Consistency

Enguerrand Prebet André Platzer

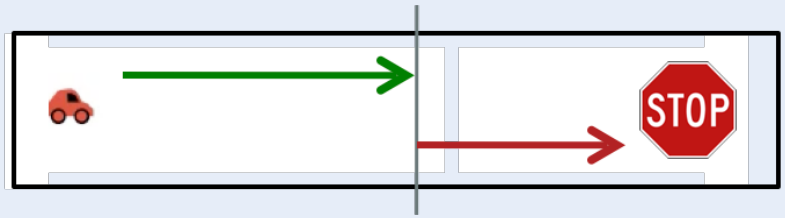
Karlsruhe Institute of Technology

IJCAR'24

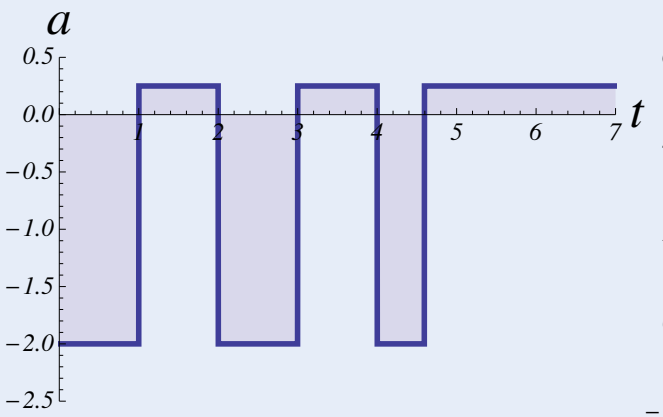
Concept (Differential Refinement Logic) (LICS'16)

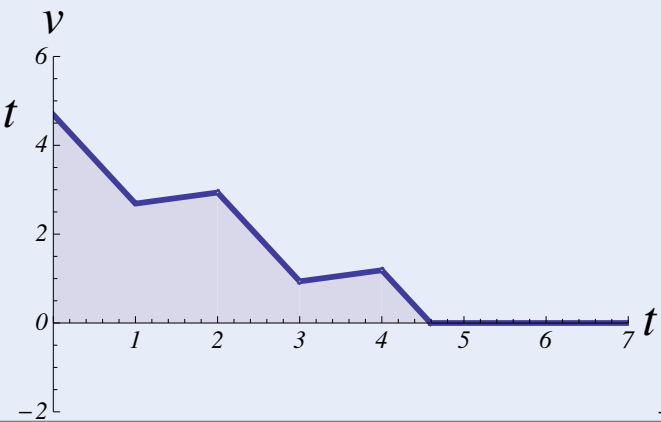


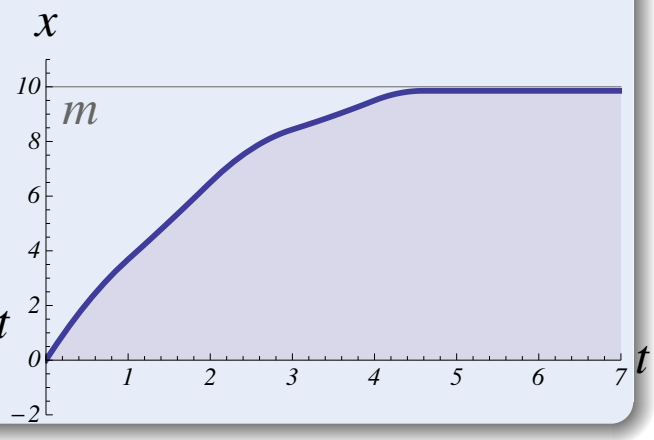
event-triggered



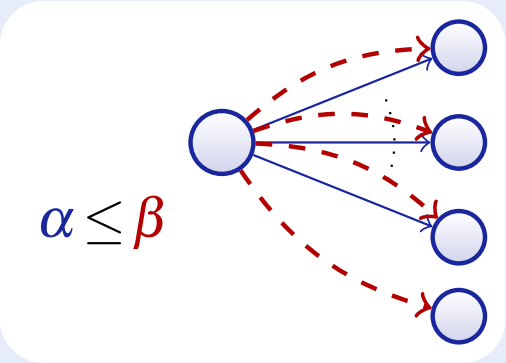
$$(u :\in G(x); x' = f(x) \,\&\, Q(x))^*$$



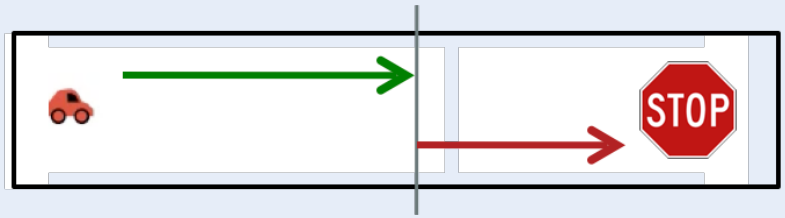




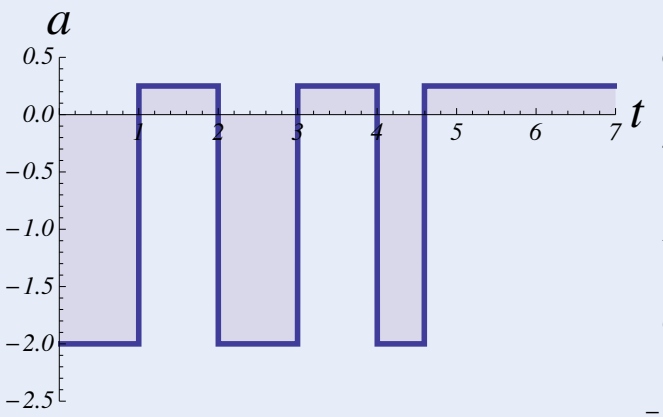
Concept (Differential Refinement Logic) (LICS'16)

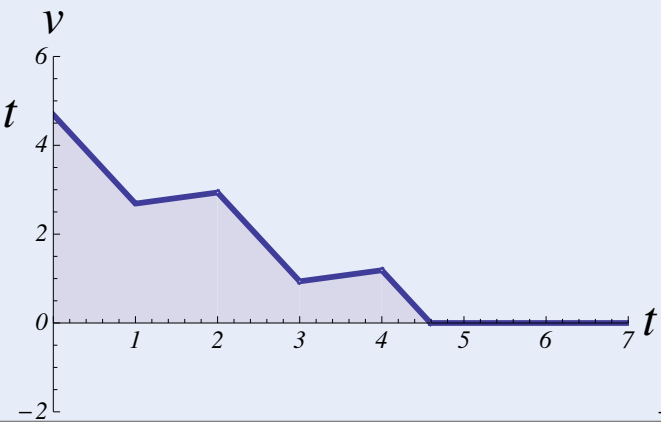


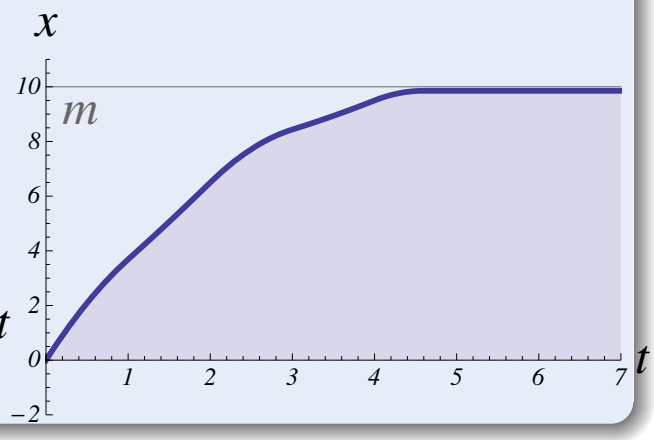
event-triggered



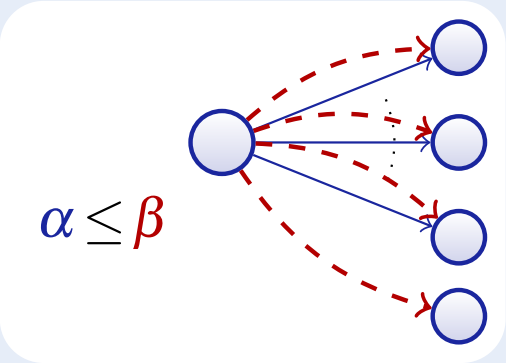
$$[(u :\in G(x); x' = f(x) \,\&\, Q(x))^*]\text{safe}$$



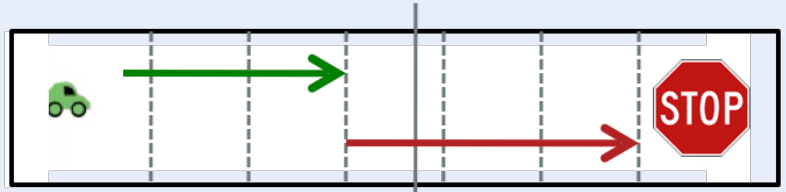




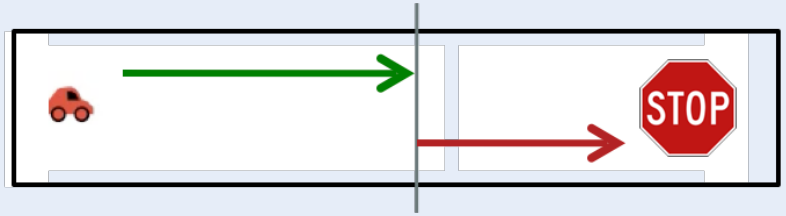
Concept (Differential Refinement Logic) (LICS'16)



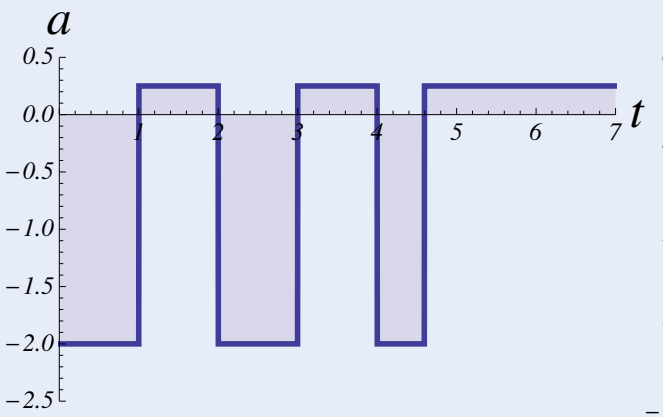
time-triggered

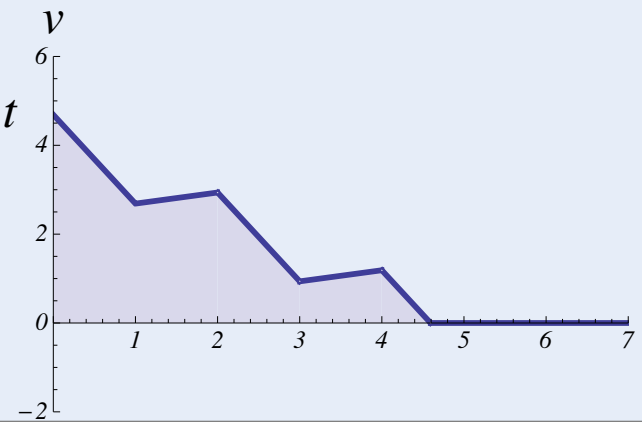


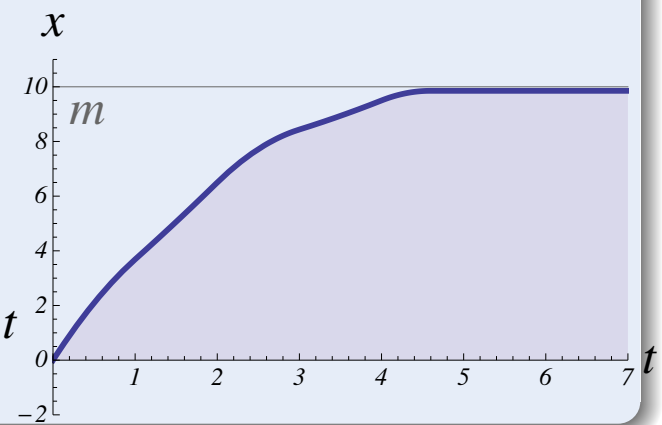
event-triggered



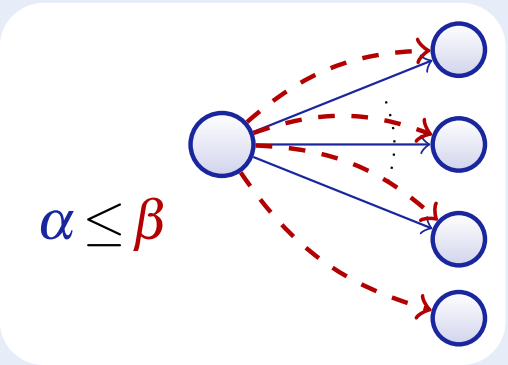
$[(u := g(x); x' = f(x) \,\&\, t \leq T)^*]\text{safe}$ $[(u :\in G(x); x' = f(x) \,\&\, Q(x))^*]\text{safe}$







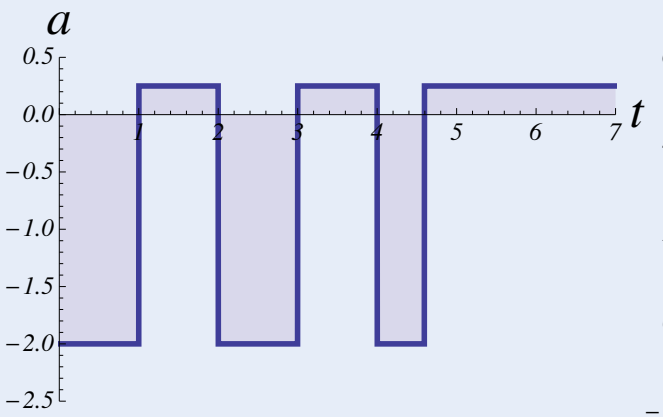
Concept (Differential Refinement Logic) (LICS'16)

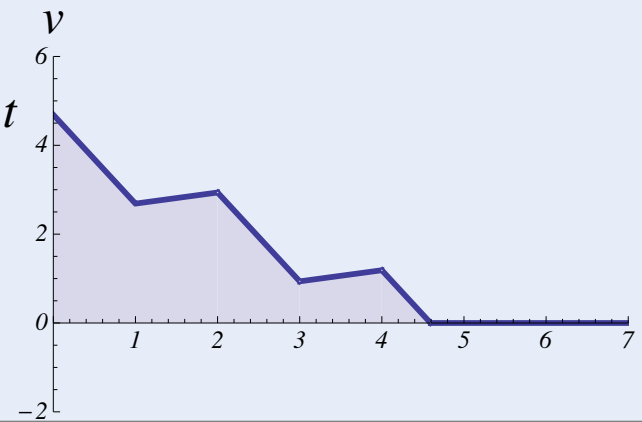


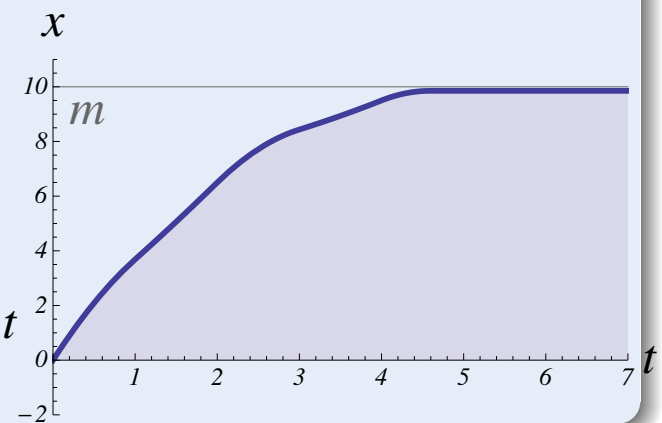
time-triggered
implementable

event-triggered
verifiable

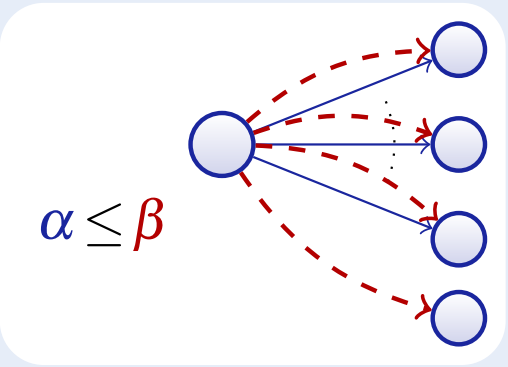
$[(u := g(x); x' = f(x) \,\&\, t \leq T)^*]\text{safe}$ $[(u :\in G(x); x' = f(x) \,\&\, Q(x))^*]\text{safe}$







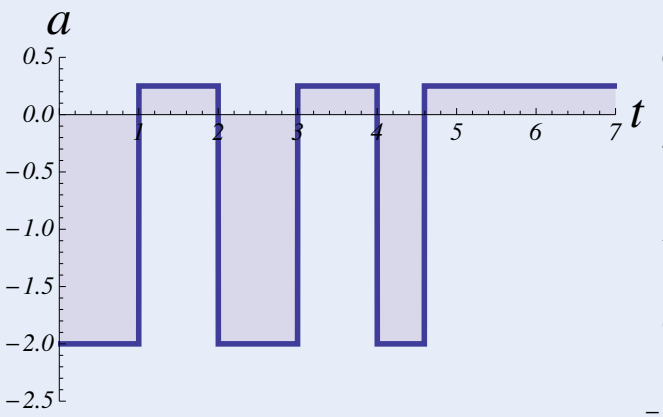
Concept (Differential Refinement Logic) (LICS'16)

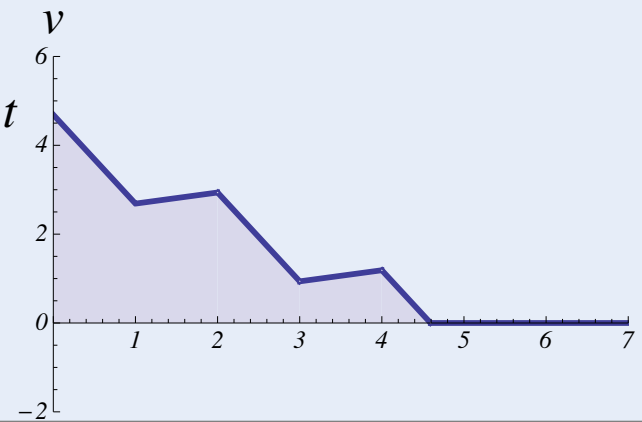


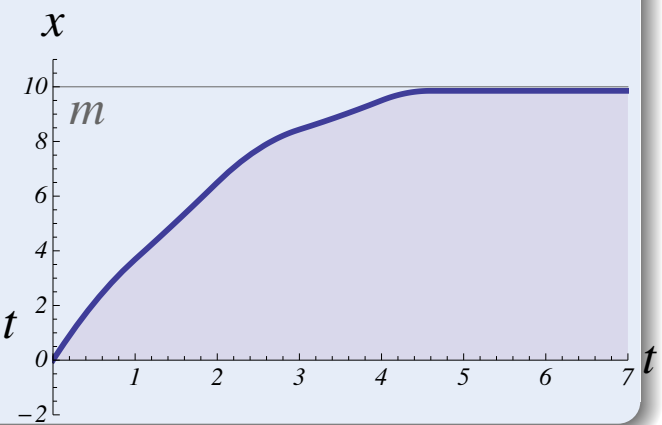
time-triggered
implementable

event-triggered
verifiable

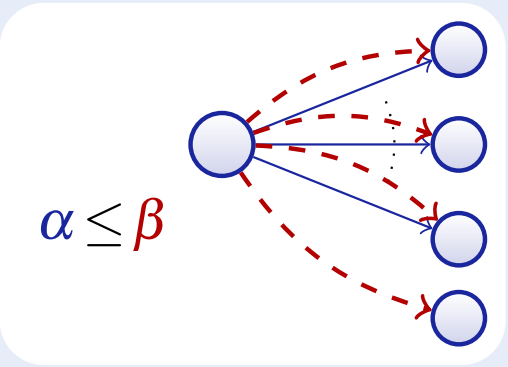
$$[(u := g(x); x' = f(x) \,\&\, t \leq T)^*]\text{safe} \leftarrow [(u :\in G(x); x' = f(x) \,\&\, Q(x))^*]\text{safe}$$







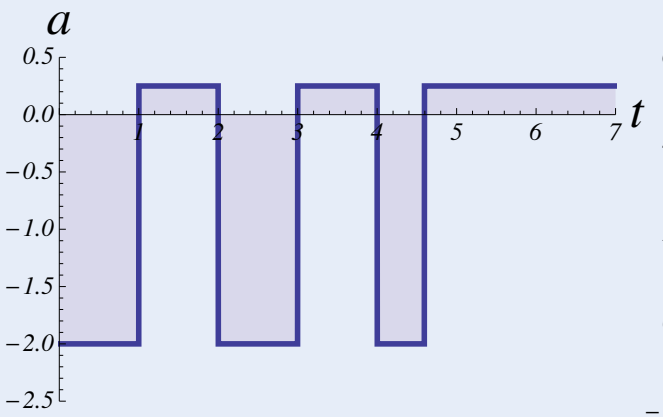
Concept (Differential Refinement Logic) (LICS'16)

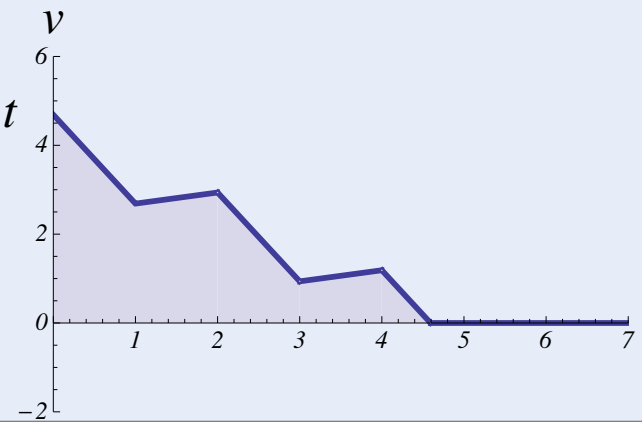


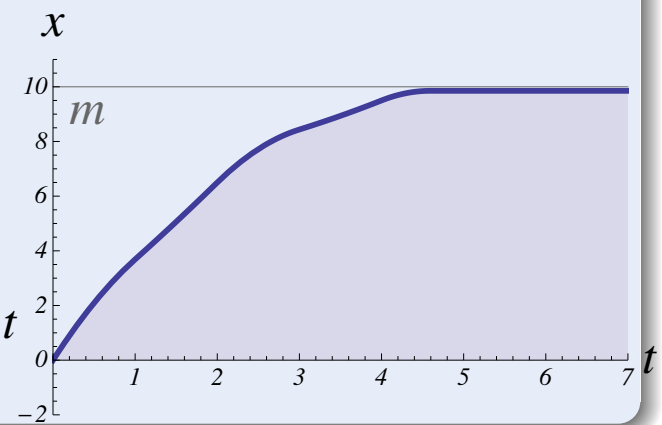
time-triggered
implementable

event-triggered
verifiable

$$(u := g(x); x' = f(x) \,\&\, t \leq T)^* \quad \leq \quad (u :\in G(x); x' = f(x) \,\&\, Q(x))^*$$



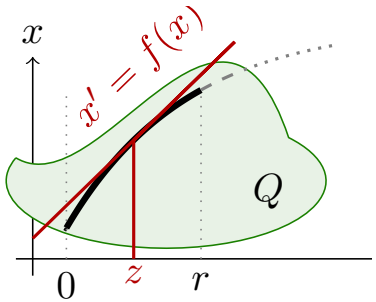


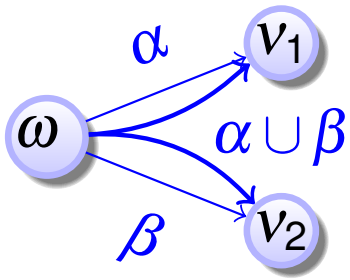


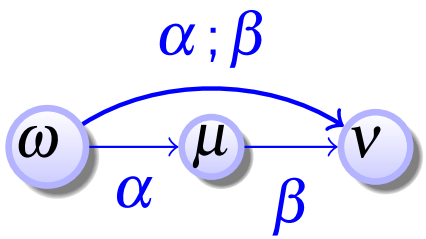
Differential Refinement Logic dRL

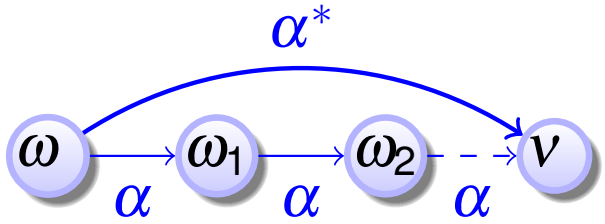
Definition (Hybrid program)

$$\alpha, \beta \ ::= \ x := e \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$



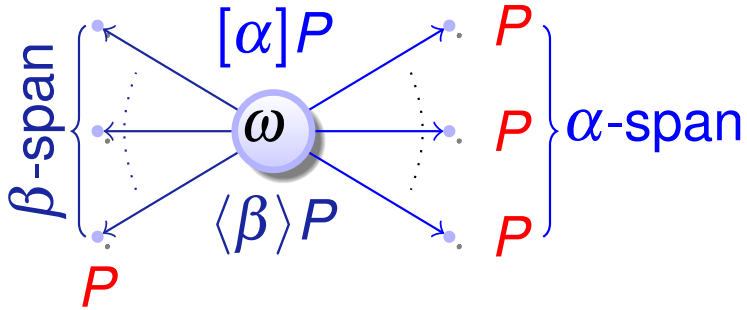




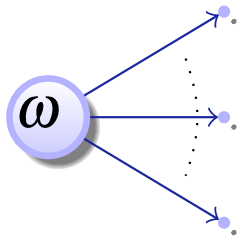


Definition (Differential refinement logic) (LICS'16)

$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid \ \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P \mid \alpha \leq \beta$$



α

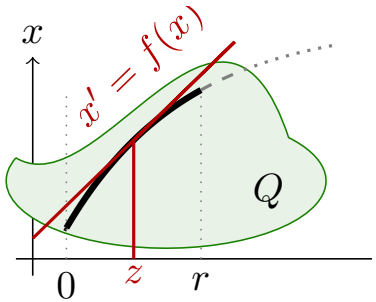


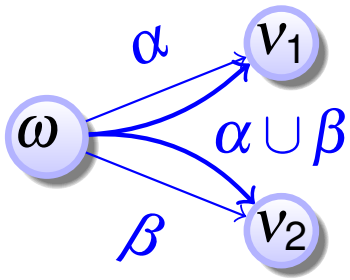
refines

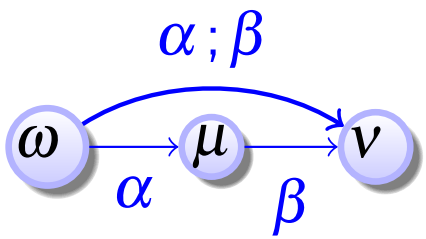
Differential Refinement Logic dRL

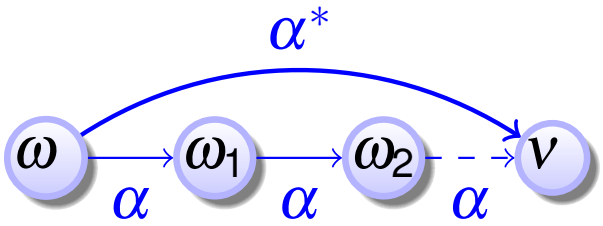
Definition (Hybrid program)

$$\alpha, \beta \ ::= \ x := e \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$



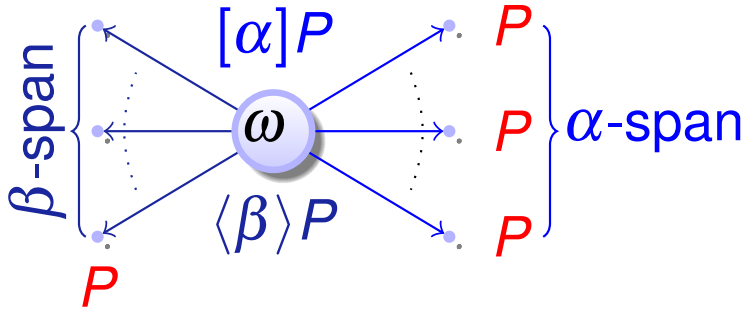


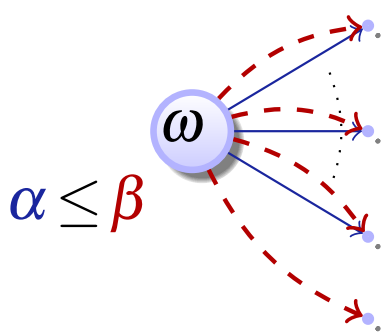




Definition (Differential refinement logic) (LICS'16)

$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid \ \mid P \to Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P \mid \alpha \leq \beta$$





refines

α less behavior

Differential Refinement Logic dRL: Semantics

Definition (Hybrid program semantics) $([\![\cdot]\!] : \mathrm{HP} \to \wp(\mathscr{S} \times \mathscr{S}))$

$$
\begin{aligned}
[\![x := e]\!] &= \{(\omega, \nu) \,:\, \nu = \omega \text{ except } \nu[\![x]\!] = \omega[\![e]\!]\} \\
[\![?Q]\!] &= \{(\omega, \omega) \,:\, \omega \in [\![Q]\!]\} \\
[\![x' = f(x)]\!] &= \{(\varphi(0), \varphi(r)) \,:\, \varphi \models x' = f(x) \text{ for some duration } r\} \\
[\![\alpha \cup \beta]\!] &= [\![\alpha]\!] \cup [\![\beta]\!] \\
[\![\alpha; \beta]\!] &= [\![\alpha]\!] \circ [\![\beta]\!] \\
[\![\alpha^*]\!] &= [\![\alpha]\!]^* = \textstyle\bigcup_{n \in \mathbb{N}} [\![\alpha^n]\!]
\end{aligned}
$$

compositional semantics

Definition (dRL semantics) $([\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathscr{S}))$

$$
\begin{aligned}
[\![\alpha \leq \beta]\!] &= \{\omega \,:\, \{\nu \,:\, (\omega, \nu) \in [\![\alpha]\!]\} \subseteq \{\nu \,:\, (\omega, \nu) \in [\![\beta]\!]\}\} \\
[\![e \geq \tilde{e}]\!] &= \{\omega \,:\, \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= [\![P]\!]^{\complement} \\
[\![P \wedge Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![\langle \alpha \rangle P]\!] &= [\![\alpha]\!] \circ [\![P]\!] = \{\omega \,:\, \nu \in [\![P]\!] \text{ for some } \nu \,:\, (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![[\alpha] P]\!] &= [\![\neg \langle \alpha \rangle \neg P]\!] = \{\omega \,:\, \nu \in [\![P]\!] \text{ for all } \nu \,:\, (\omega, \nu) \in [\![\alpha]\!]\}
\end{aligned}
$$

Differential Refinement Logic: Axiomatization

$[\leq]$ $\alpha \leq \beta \to ([\alpha]P \leftarrow [\beta]P)$

$\langle\leq\rangle$ $\beta \leq \alpha \to (\langle\alpha\rangle P \leftarrow \langle\beta\rangle P)$

; $\alpha;\beta \leq \gamma;\delta \leftarrow \alpha \leq \gamma \wedge [\alpha]\beta \leq \delta$

un$*$ $\alpha^* \leq \beta^* \leftarrow [\alpha^*](\alpha \leq \beta)$

loop$_l$ $\alpha^*;\beta \leq \beta \leftarrow [\alpha^*]\alpha;\beta \leq \beta$

loop$_r$ $\alpha;\beta^* \leq \alpha \leftarrow \alpha;\beta \leq \alpha$

ODE $x' = e \,\&\, P \leq x' = k \,\&\, Q \leftrightarrow [x' = e \,\&\, P](x' = k \wedge Q)$

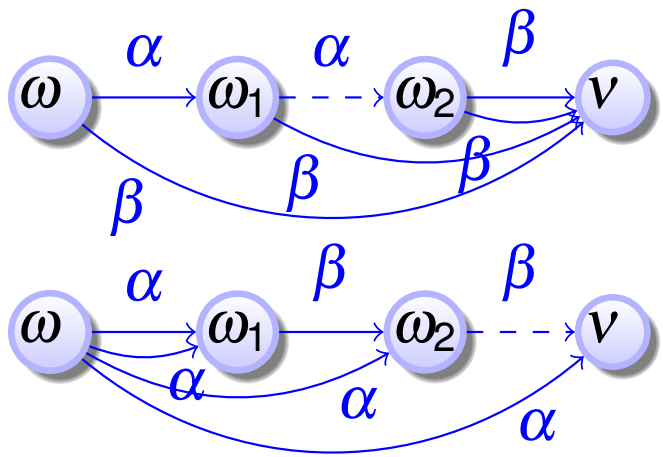
$\cup_l$ $\alpha \cup \beta \leq \gamma \leftrightarrow \alpha \leq \gamma \wedge \beta \leq \gamma$

$\cup_r$ $\alpha \leq \beta \cup \gamma \leftarrow \alpha \leq \beta \vee \alpha \leq \gamma$

$\leq_t$ $\alpha \leq \beta \leftarrow \alpha \leq \gamma \wedge \gamma \leq \beta$

$\leq$ $\alpha \leq \beta \leftrightarrow \forall y\,(\langle\alpha\rangle x = y \to \langle\beta\rangle x = y)$

$\leq'$ $[\alpha]P \leftrightarrow \alpha \leq (x := *; ?P)$



LICS'16,IJCAR'24

Differential Refinement Logic: Axiomatization

$[\leq]$ $\alpha \leq \beta \to ([\alpha]P \leftarrow [\beta]P)$

$\langle\leq\rangle$ $\beta \leq \alpha \to (\langle\alpha\rangle P \leftarrow \langle\beta\rangle P)$

; $\alpha;\beta \leq \gamma;\delta \leftarrow \alpha \leq \gamma \wedge [\alpha]\beta \leq \delta$

un$*$ $\alpha^* \leq \beta^* \leftarrow [\alpha^*](\alpha \leq \beta)$

loop_l $\alpha^*;\beta \leq \beta \leftarrow [\alpha^*]\alpha;\beta \leq \beta$

loop_r $\alpha;\beta^* \leq \alpha \leftarrow \alpha;\beta \leq \alpha$

ODE $x' = e \,\&\, P \leq x' = k \,\&\, Q \leftrightarrow [x' = e \,\&\, P](x' = k \wedge Q)$

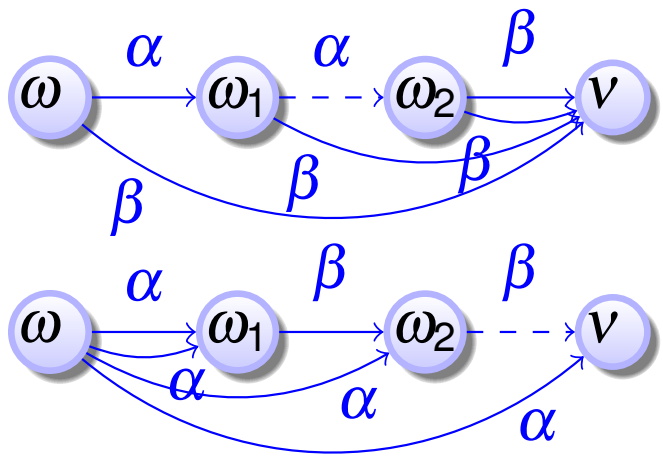
$\cup_l$ $\alpha \cup \beta \leq \gamma \leftrightarrow \alpha \leq \gamma \wedge \beta \leq \gamma$

$\cup_r$ $\alpha \leq \beta \cup \gamma \leftarrow \alpha \leq \beta \vee \alpha \leq \gamma$

$\leq_t$ $\alpha \leq \beta \leftarrow \alpha \leq \gamma \wedge \gamma \leq \beta$

Property via refine

Refine via property



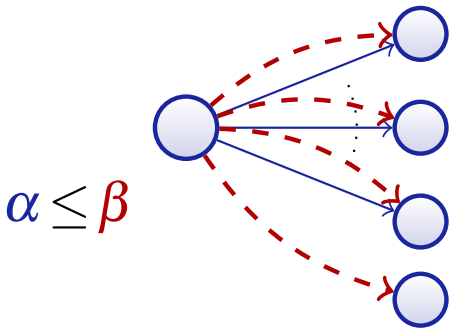
LICS'16,IJCAR'24

Takeaway: Hybrid System Refinements for Concistency

Differential refinement logic

- Event-triggered control: Easy to verify but hard to implement
- Time-triggered control: Easy to implement but hard to verify
- Best of both worlds: verify event-triggered, implement time-triggered
- dRL proofs identify required conditions (e.g., event invariance)
- Implementation model $\neq$ verification model But consistency!
- Iterative design reduces risk, increases repeated effort
- Hierarchical proof structuring by refinement
- Decidable fragment for refinements via equational ODEs JACM'20

Relations $\alpha \leq \beta$ between hybrid systems models are as useful as properties $[\alpha]\varphi$ of hybrid systems models. Fundamental consistency operator.
Simultaneous logical language integration is best.



LICS'16,IJCAR'24

GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft – SFB-1608 – 501798263

CONVIDE CRC 1608

THANK YOU!

TECHNISCHE UNIVERSITÄT DRESDEN

Technische Universität München

UNIVERSITY OF MANNHEIM

KIT Karlsruher Institut für Technologie

GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft

CONVIDE CRC 1608

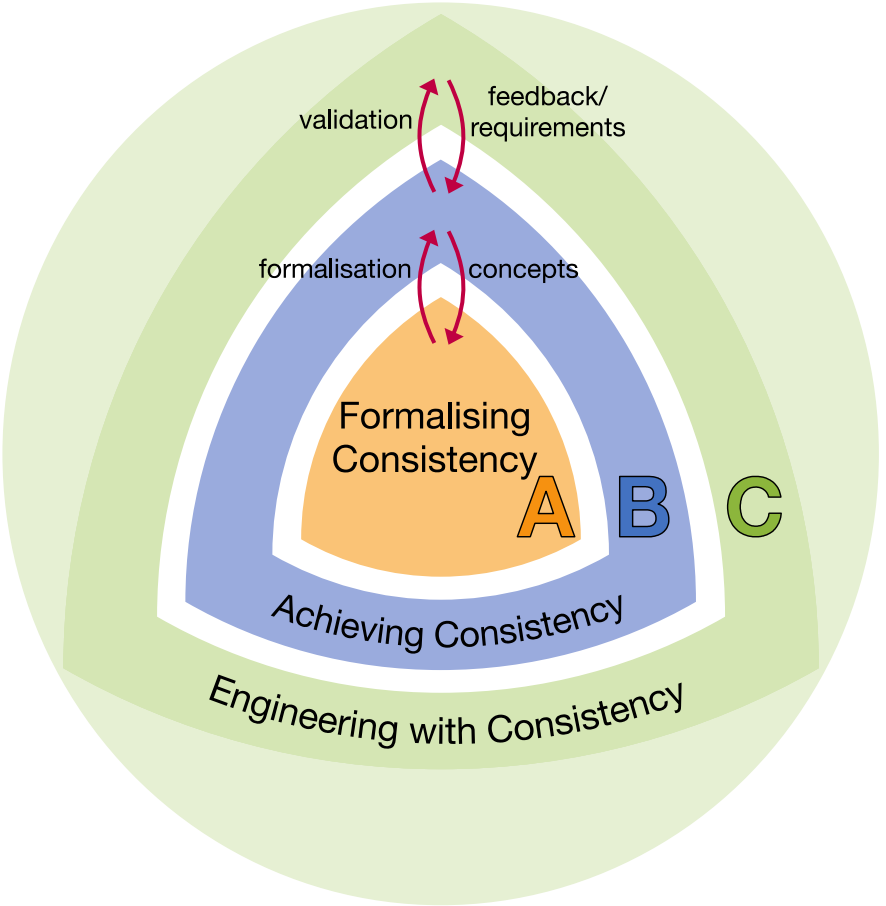
CONVIDE
RESEARCH AREA B

Prof. Dr. Ina Schaefer • Overview • 12.12.2024

ROLE OF RESEARCH AREA B WITHIN CONVIDE

Mechanisms for consistency management across different views, in particular:

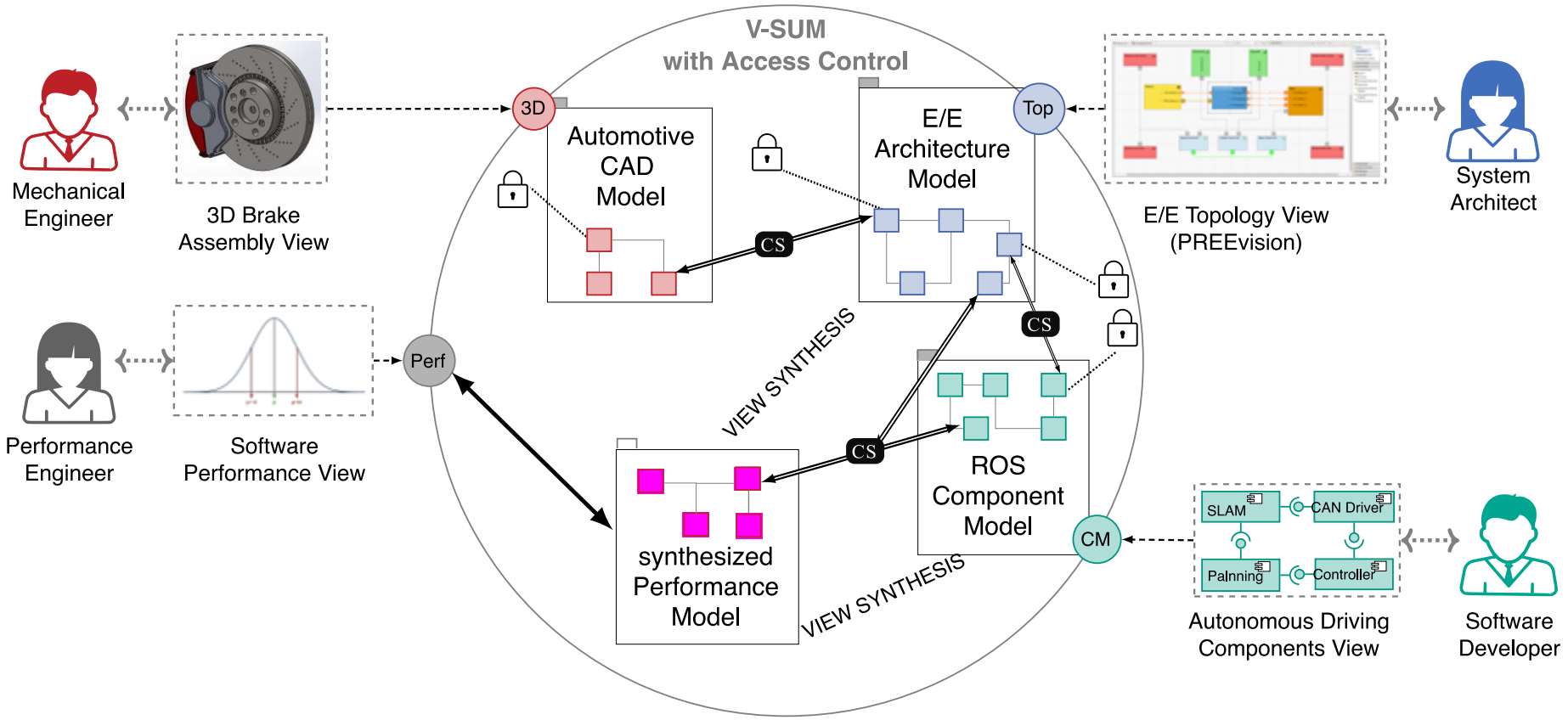
- Designing V-SUM metamodels and view types compatible with intellectual property protection
- Working concurrently on different views of a V-SUM
- Dealing with temporary inconsistencies in a V-SUM
- Maintaining consistency among variants and versions in a V-SUM



PROJECT OVERVIEW

B01	Cross-Organizational Design of View Types and V-SUM Meta-Models	Atkinson **Pretschner**	
B02	Concurrent Editing and Transactionality	Acosta **Reussner**	
B03	Recovery from Temporary Inconsistency	Koziolek **Ulbrich**	
B04	Maintaining Consistency between Variants and Versions	**Aßmann** Burger Schaefer	

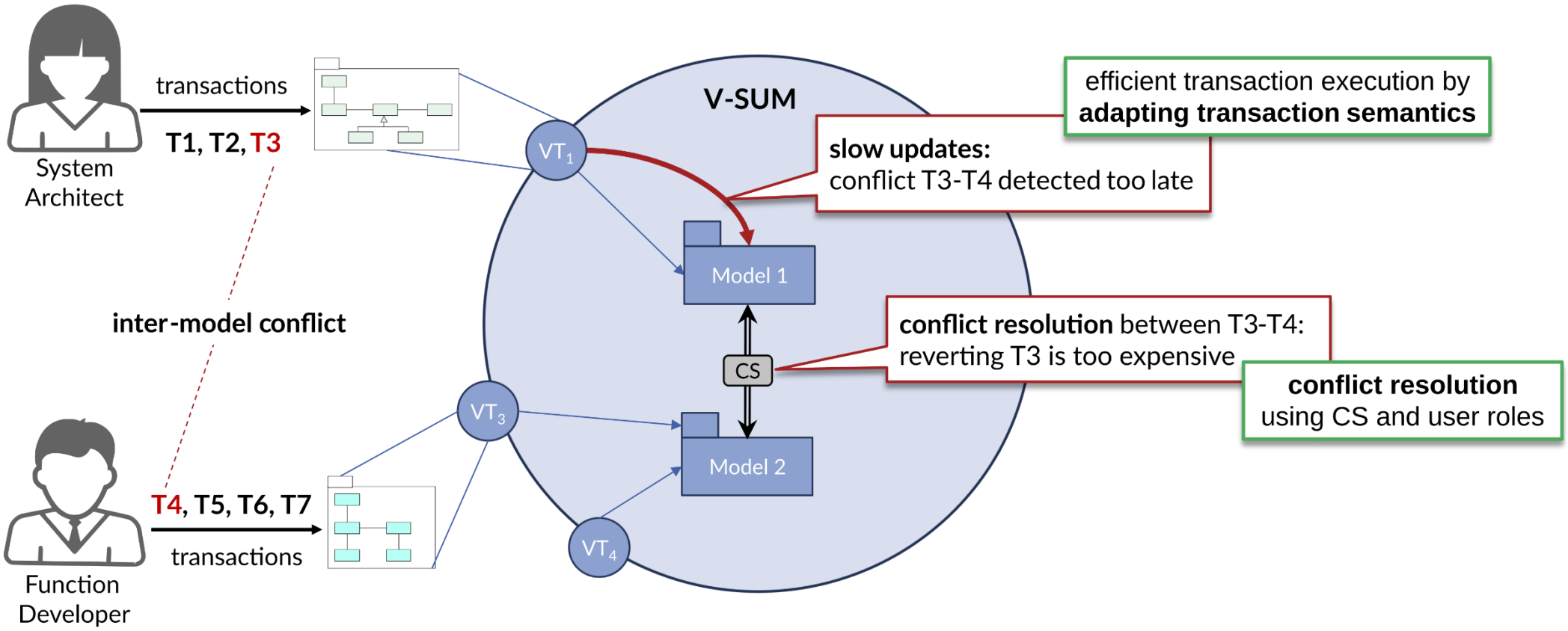
B01: CROSS-ORGANIZATIONAL DESIGN OF VIEW TYPES AND V-SUM META MODELS



Exemplary Research Question

How can V-SUM meta models and view types be defined to preserve consistency and to protect intellectual property?

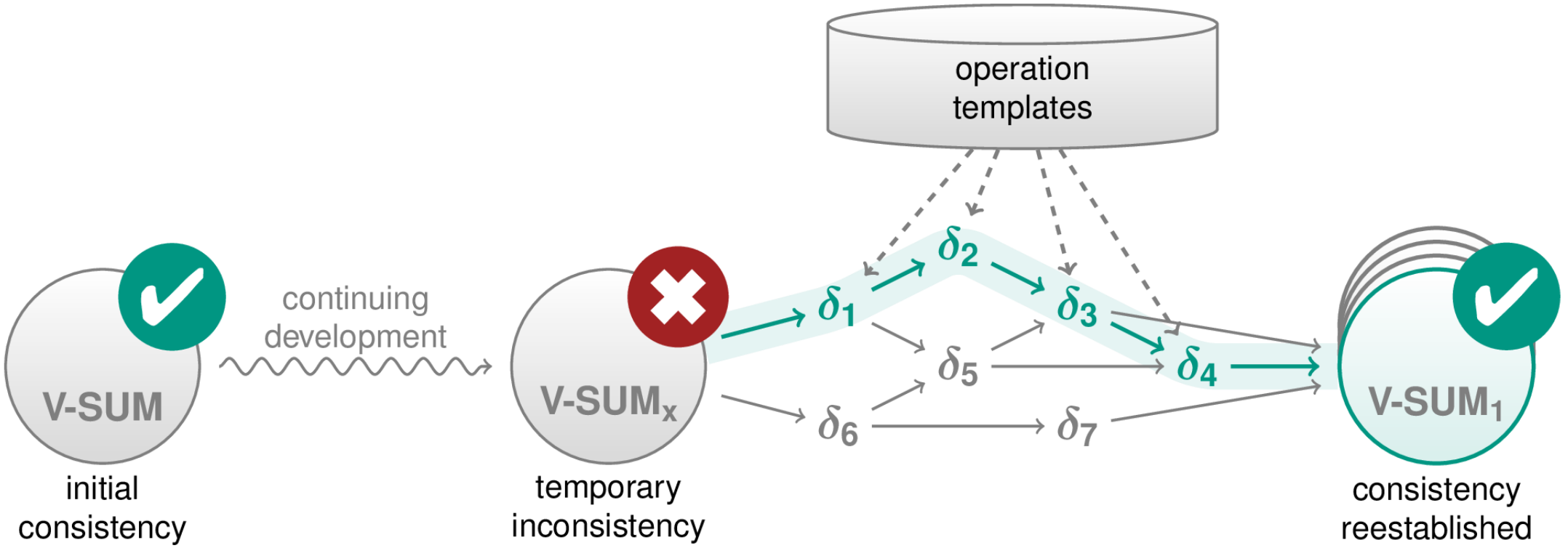
B02: CONCURRENT EDITING AND TRANSACTIONALITY



Exemplary Research Question

How can transactionality be used efficiently to preserve consistency after concurrent edits?

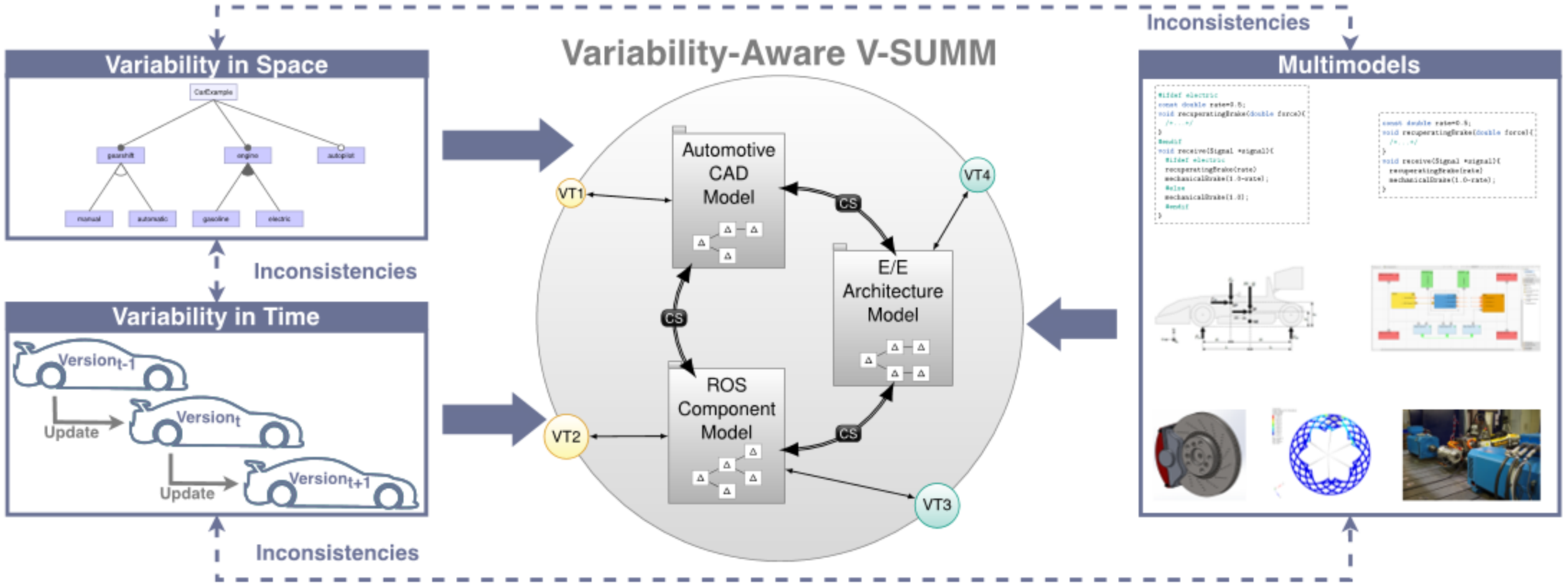
B03: RECOVERY FROM TEMPORARY INCONSISTENCY



Exemplary Research Question

How can abstract recovery operations leading out of a temporary model inconsistency be represented and efficiently searched for?

B04: MAINTAINING CONSISTENCY BETWEEN VARIANTS AND VERSIONS



Exemplary Research Question

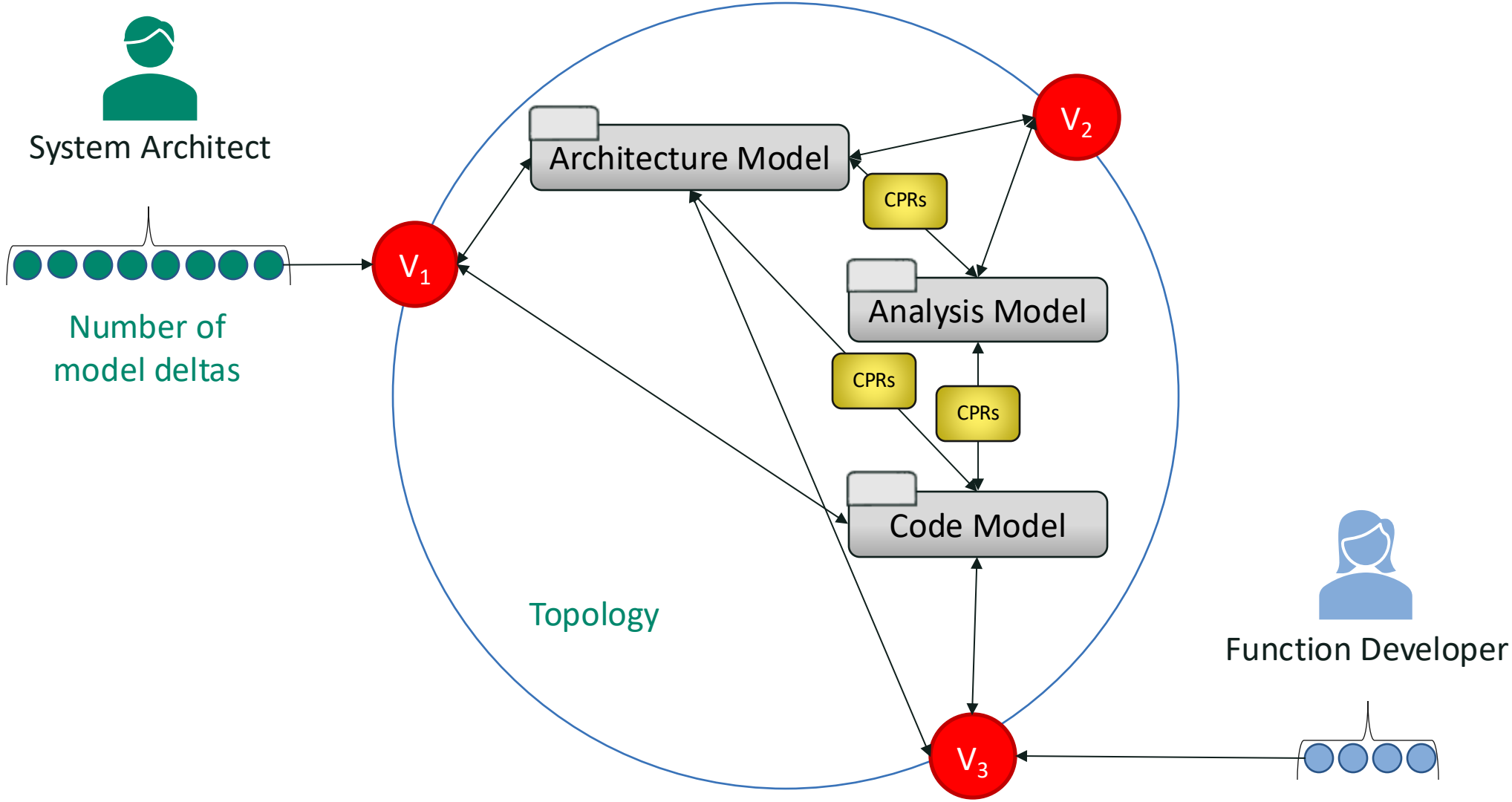
How can a V-SUM capture consistency of variants and versions and support their view-based development?

CONVIDE
CRC 1608

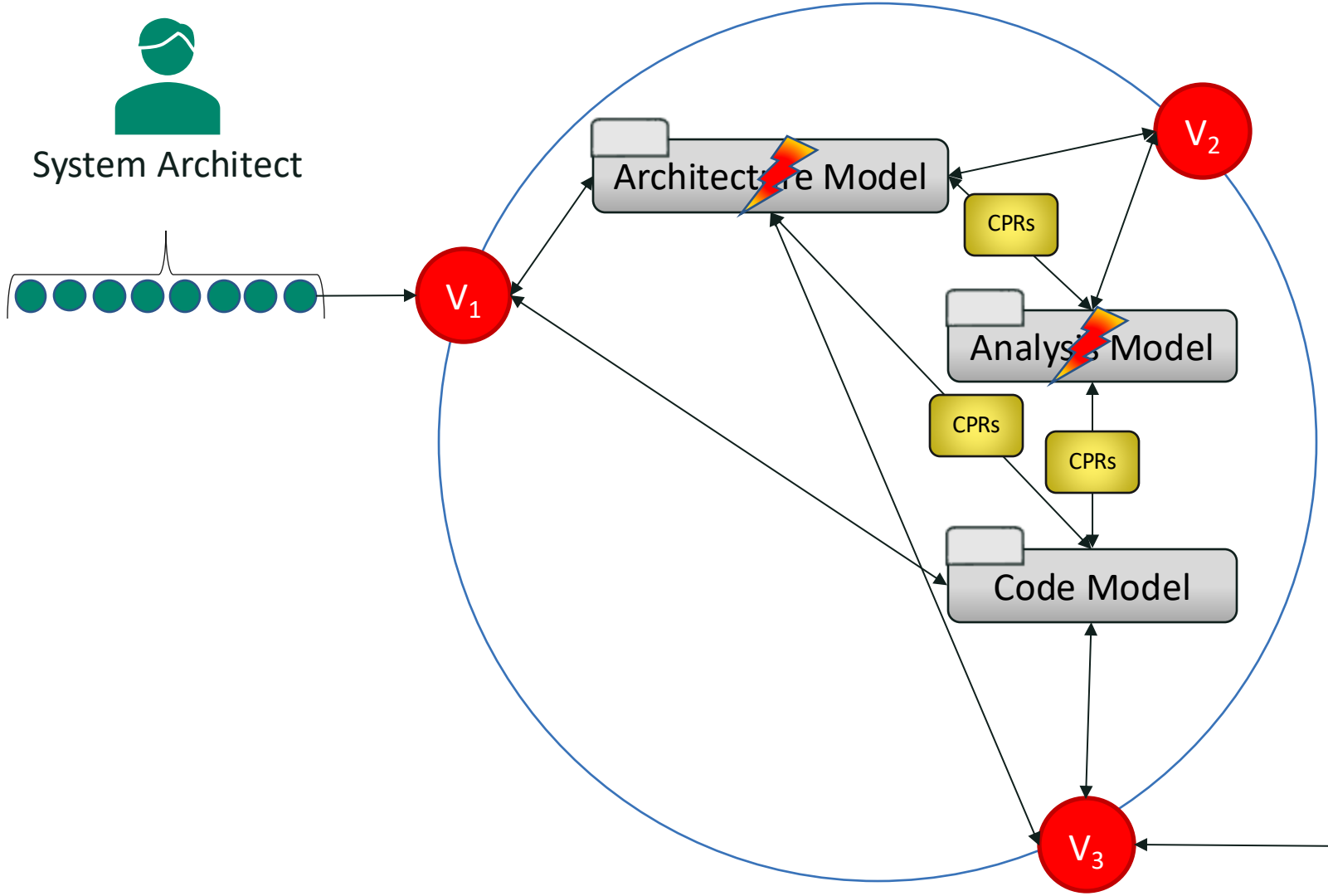
B02 @ CRC 1608

Concurrent Editing and Transactionality

PERFORMANCE OF MODEL UPDATES



CONFLICTS AND PERFORMANCE PROBLEMS



How long does it take to apply model deltas?

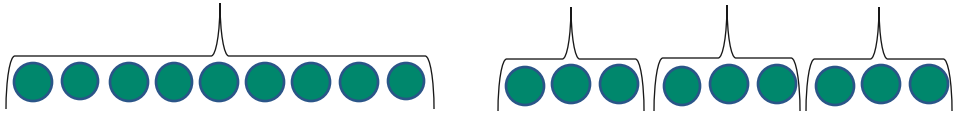
What if two transactions conflict and one needs to be rejected?

OUR APPROACH

Varying the Transaction Size

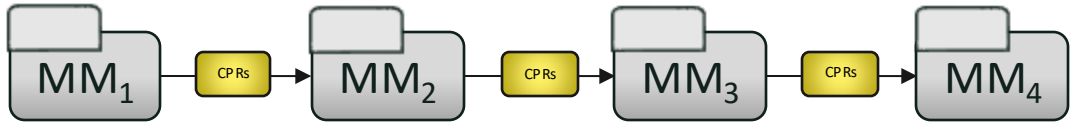
One transaction, different number of model deltas [1]

Different number of transactions, same number of model deltas [2]

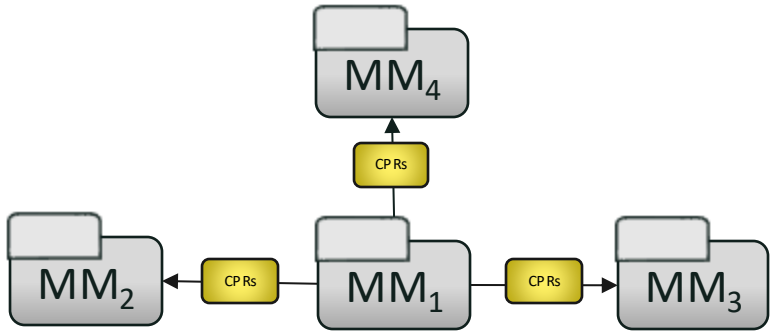


Varying the V-SUMM topology

- Vary chain length [2]



- Vary fan-out degree [2]



References

[1] Benedikt Jutz and Thomas Weber. Scalability of Consistency Preservation in Vitruvius. 15th Symposium on Software Performance, Linz, 2024.

[2] Thomas Weber, Benedikt Jutz and Zenon Zacouris. The Influence of Granularity of Transactions on Performance In Vitruvius. 15th Symposium on Software Performance, Linz, 2024.

PROCEDURE AND RESULTS

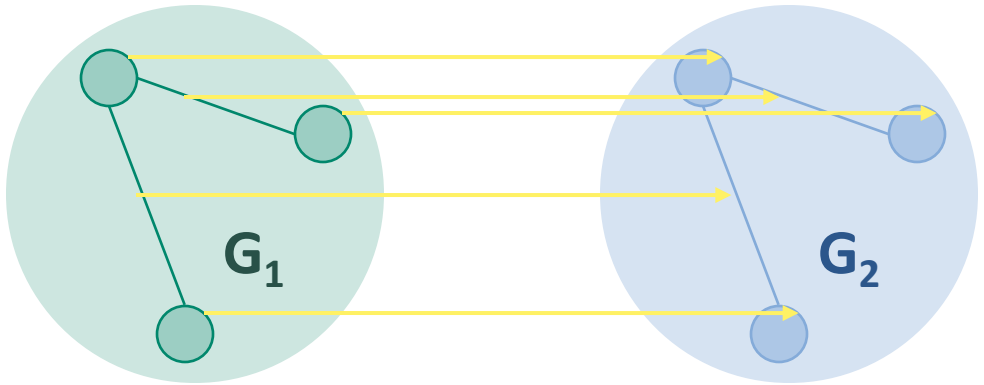
Procedure

Use two different V-SUM metamodels for measurements:

1. UML-Java case study [1]
2. Connected graph with isomorphism consistency rules [2]

Apply different types of model deltas, measure time to restore consistency:

1. UML models converted into model deltas [1]
2. Multiple nodes created in one graph [2]



Findings

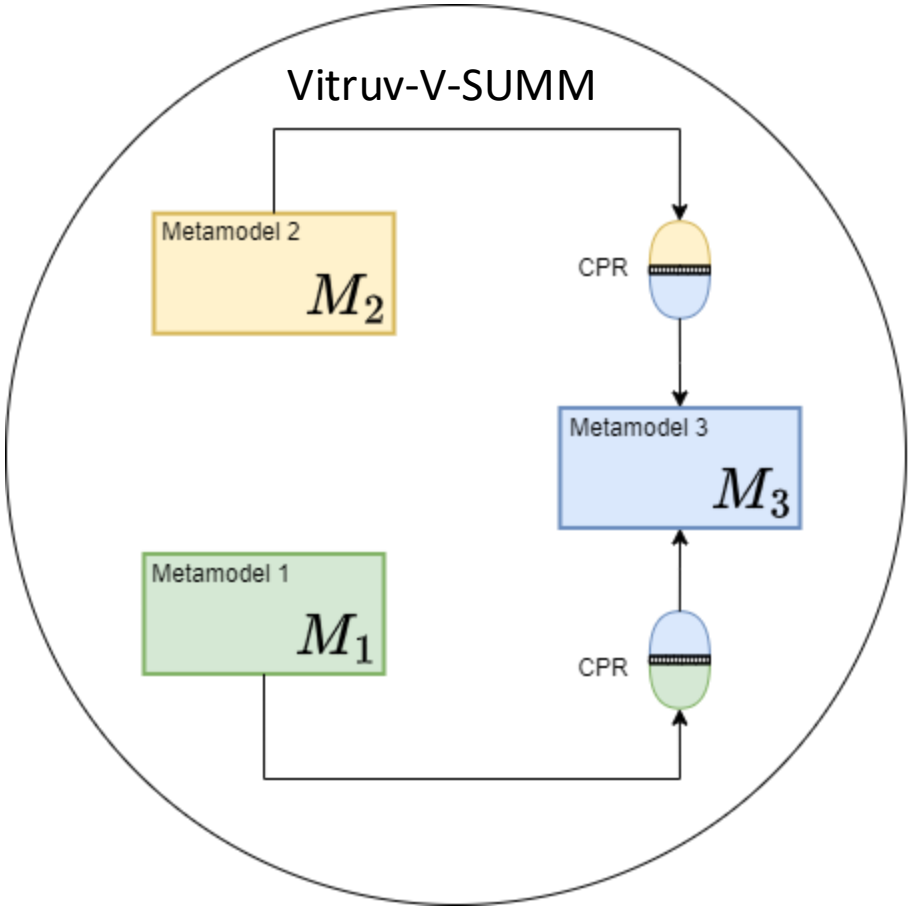
1. Time for applying consistency rules and number of model deltas correspond [1]
2. Performance degrades with smaller transaction size [2]
3. Propagating deltas is slower in a fan-out than in a chain topology [2]

CONVIDE
CRC 1608

B04 @ CRC 1608

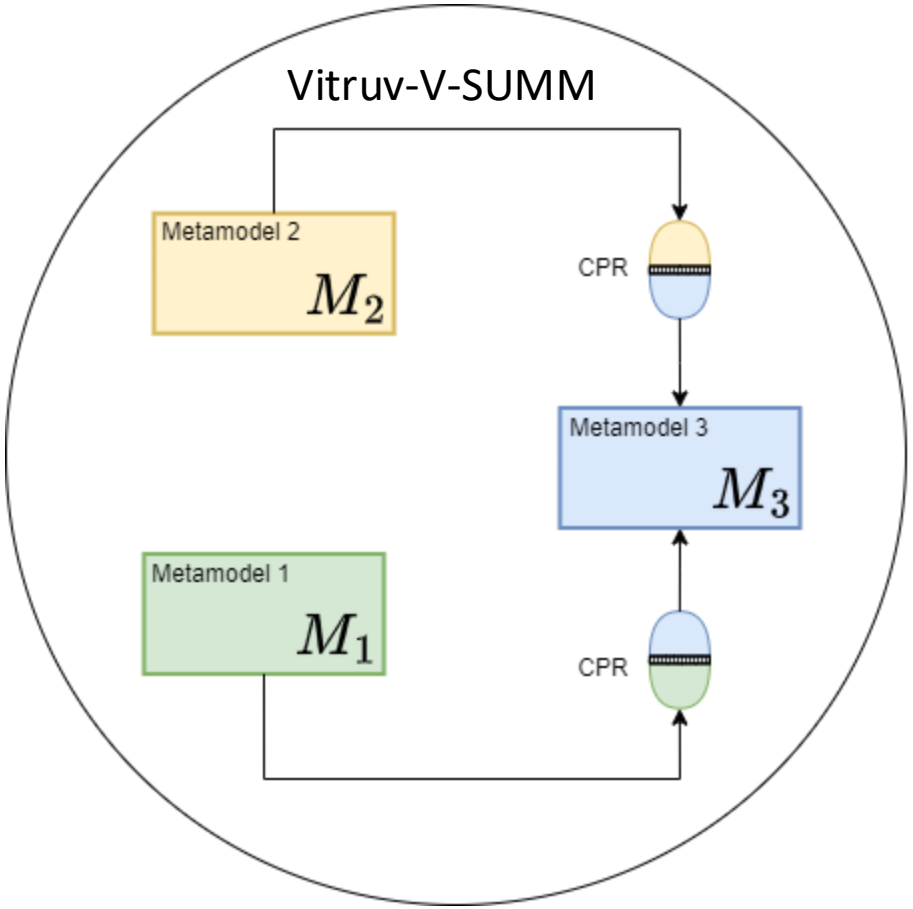
Consistency Preserving SPLE

CONSISTENCY PRESERVING SPLE – CONTEXT

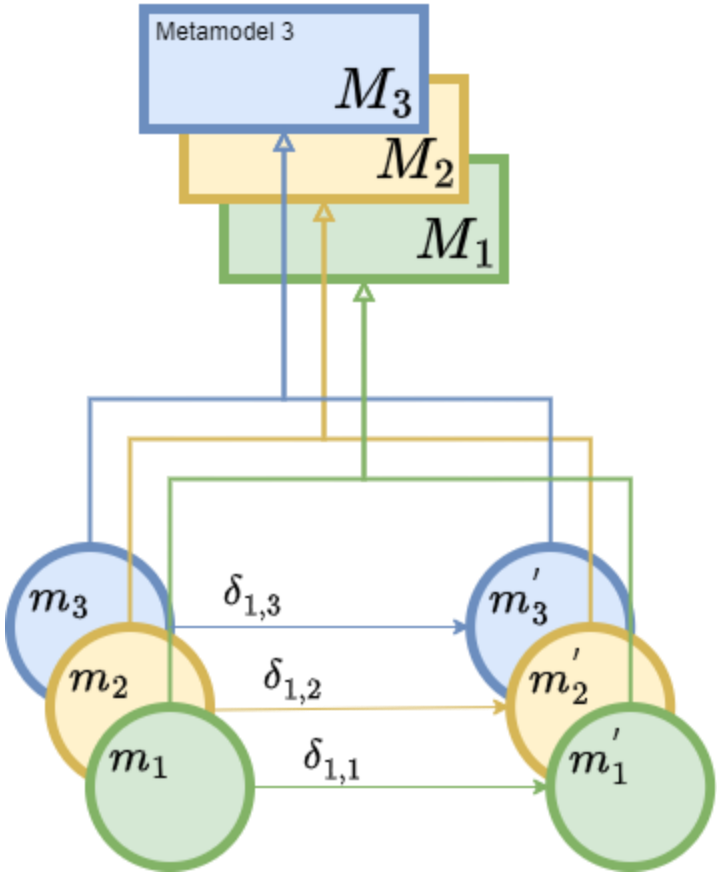


Vitruv consistency preservation in a model-driven world (unidirectional CPRs)

CONSISTENCY PRESERVING SPLE – CONTEXT

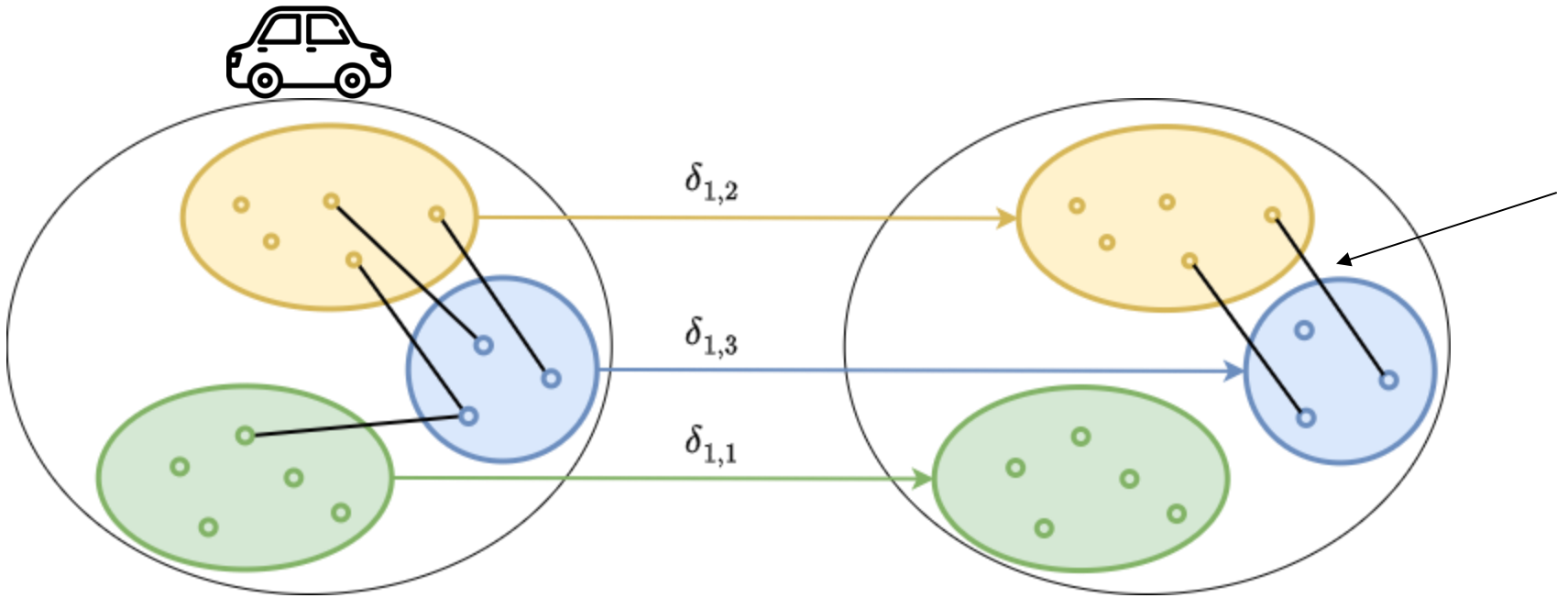


Vitruv consistency preservation in a model-driven world (unidirectional CPRs)



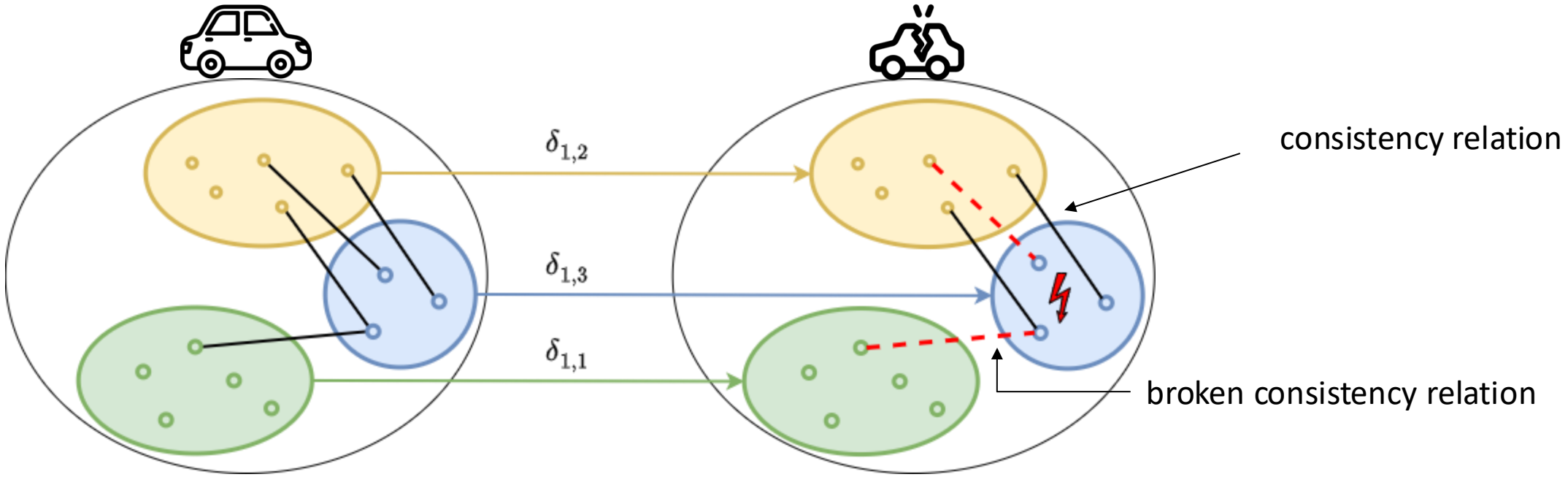
Delta-oriented variability in a model-driven world

CONSISTENCY PRESERVING SPLE – PROBLEM

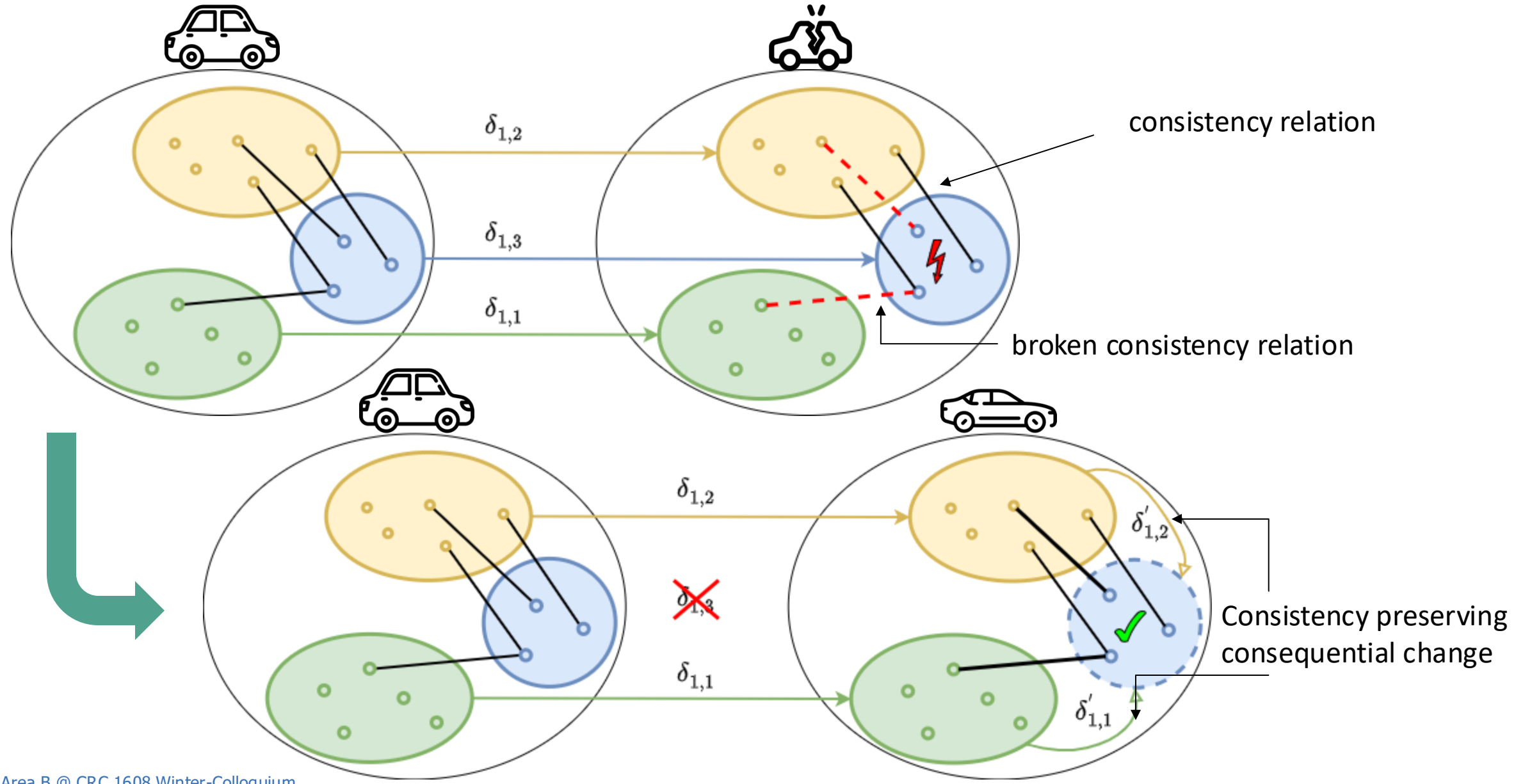


consistency relation

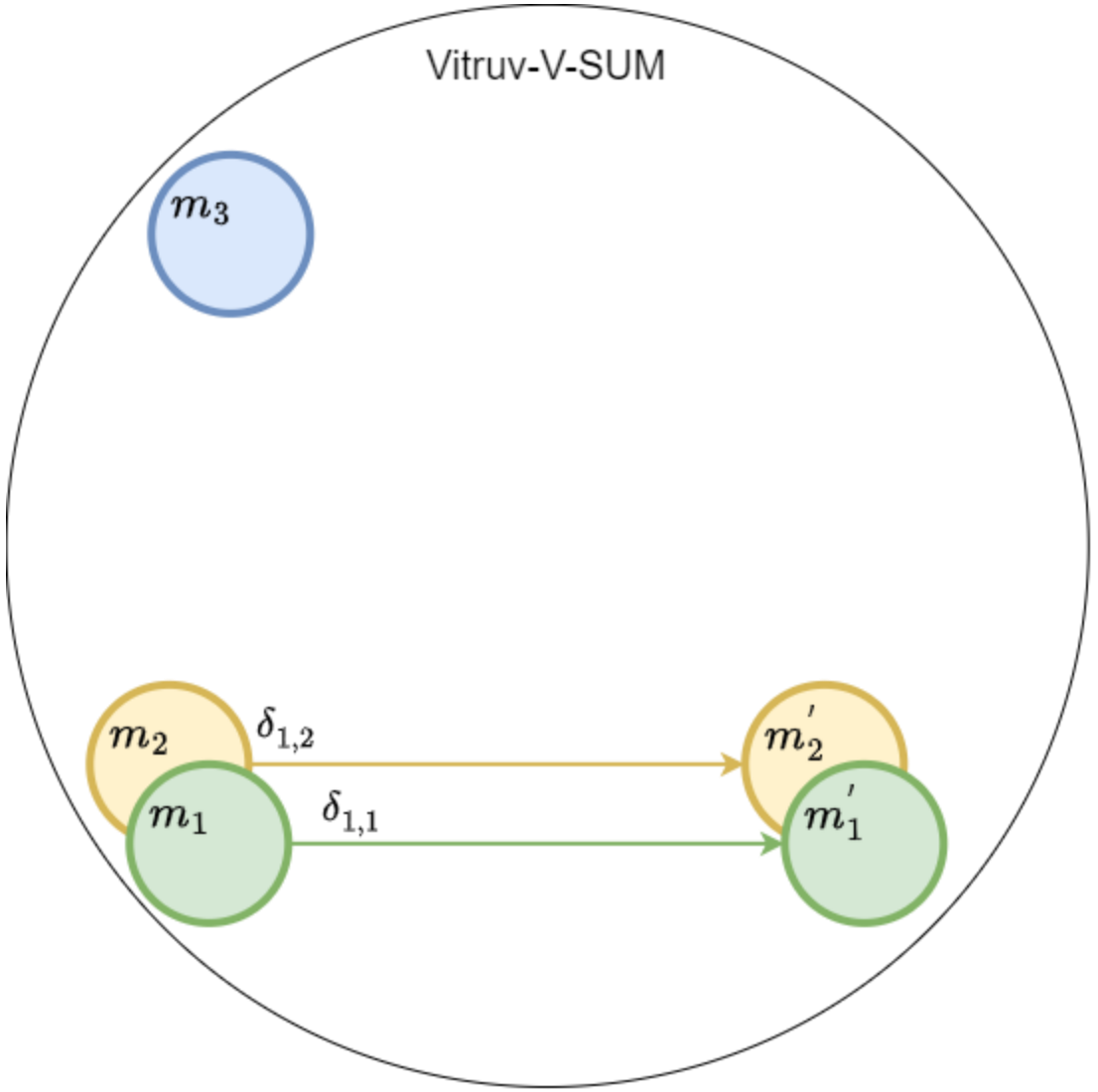
CONSISTENCY PRESERVING SPLE – PROBLEM



CONSISTENCY PRESERVING SPLE – PROBLEM



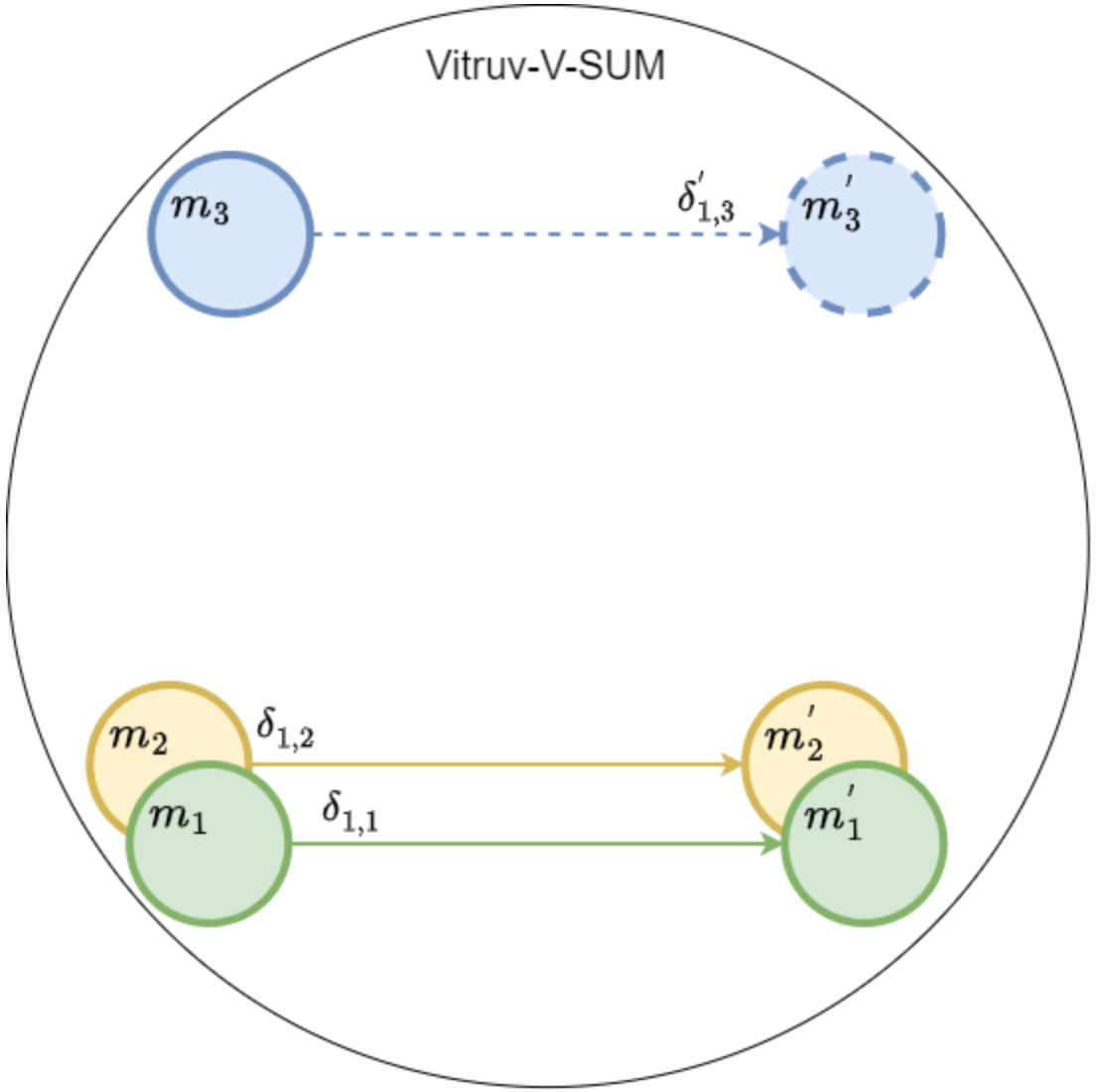
CONSISTENCY PRESERVING SPLE – IDEA



Idea:
Using consistency preservation as a mechanism for model-driven development

Advantage:
- Getting a consistent version of model 3
- Getting a delta for further SPLE development

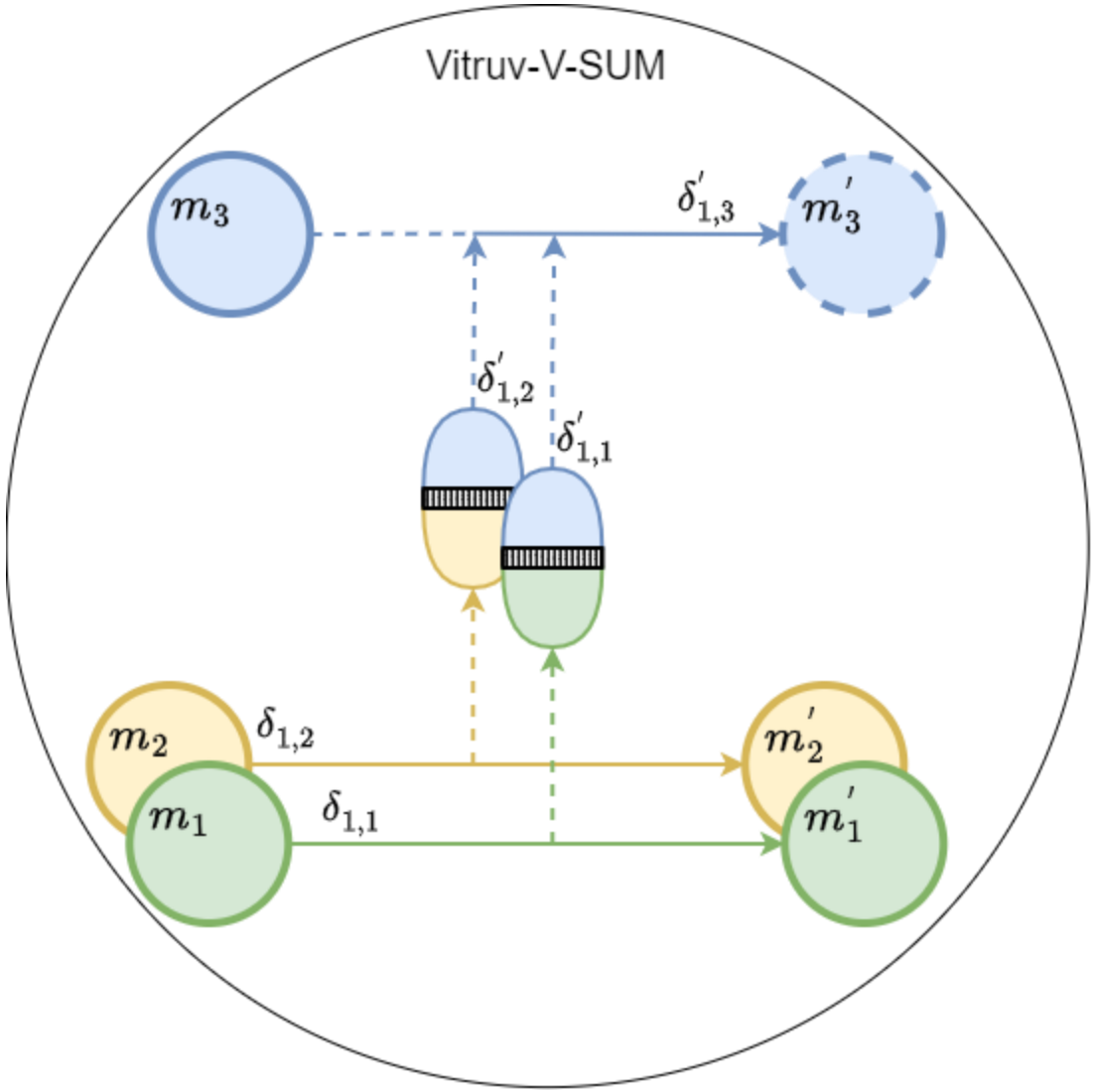
CONSISTENCY PRESERVING SPLE – IDEA



Idea:
Using consistency preservation as a mechanism for model-driven development

Advantage:
- Getting a consistent version of model 3
- Getting a delta for further SPLE development

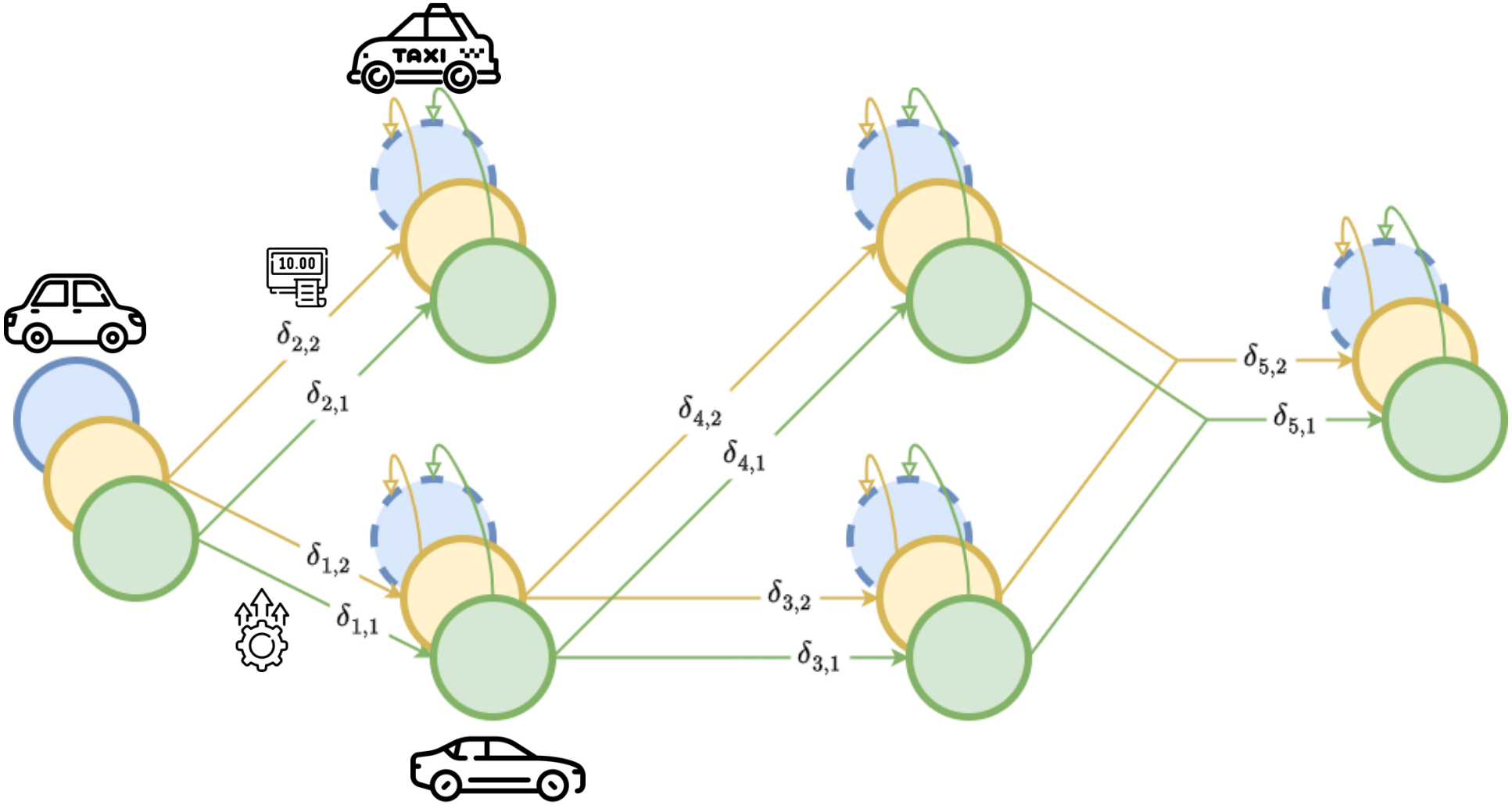
CONSISTENCY PRESERVING SPLE – IDEA



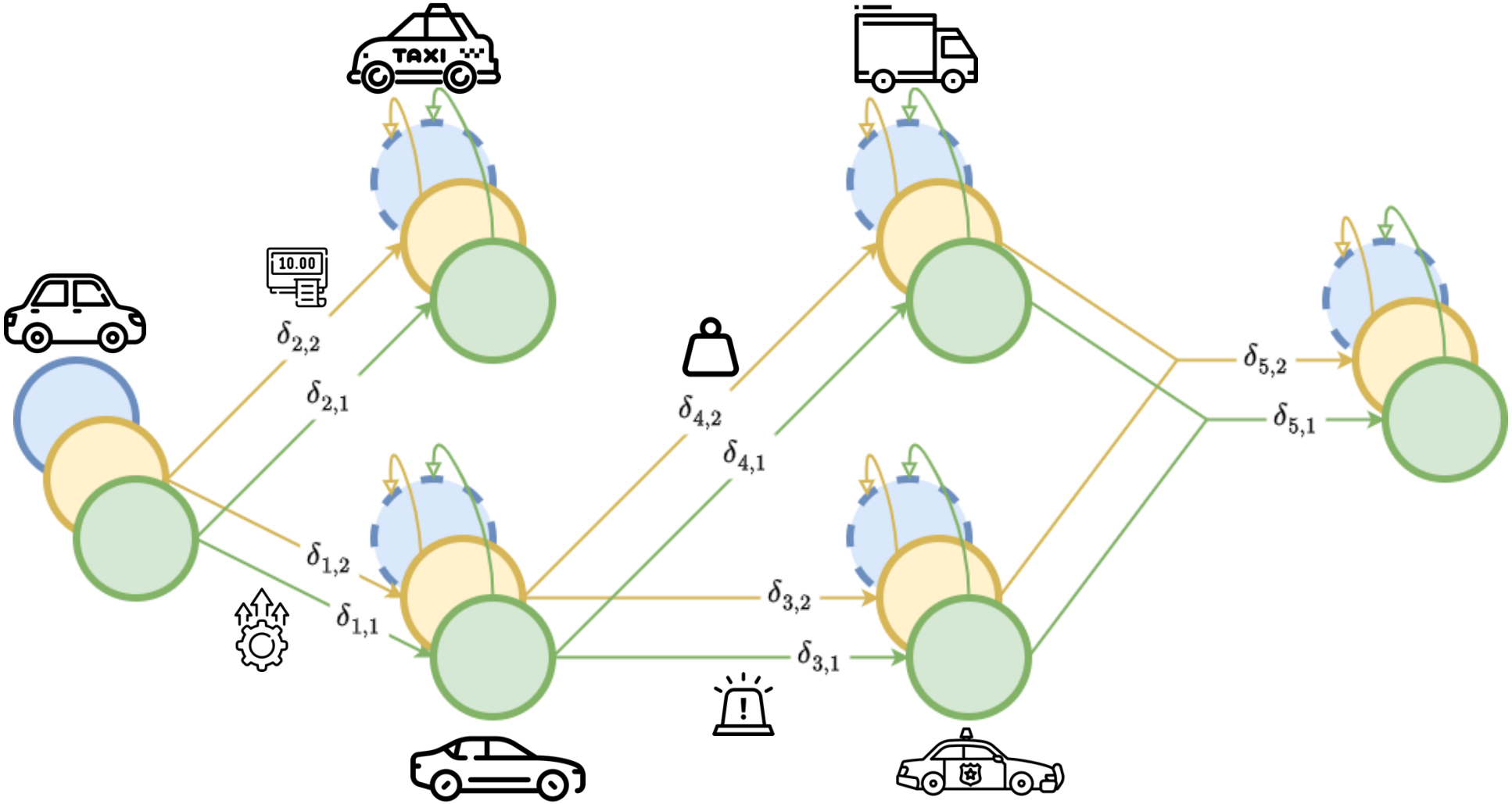
Idea:
Using consistency preservation as a mechanism for model-driven development

Advantage:
- Getting a consistent version of model 3
- Getting a delta for further SPLE development

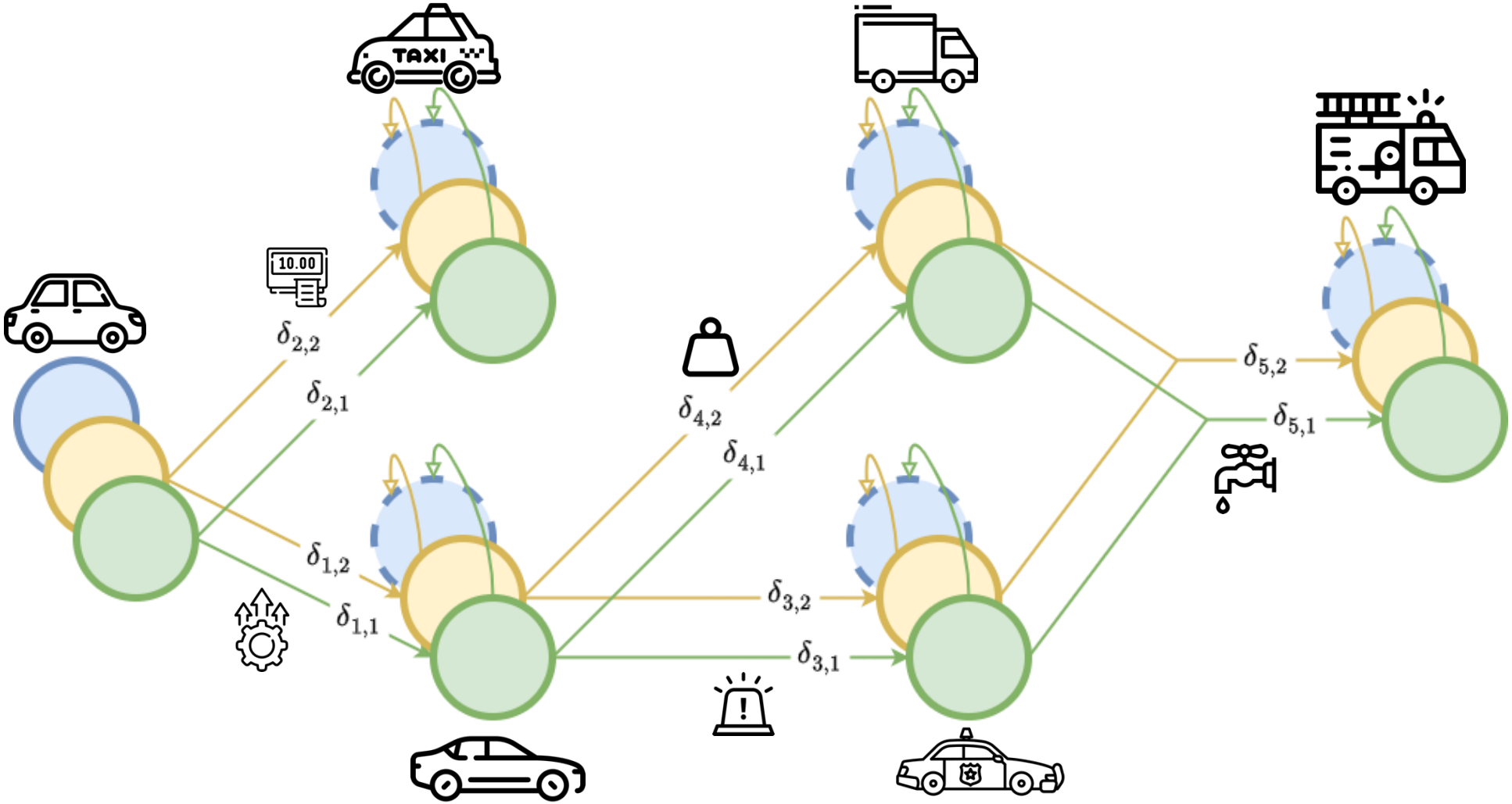
CONSISTENCY PRESERVING SPLE – VISION



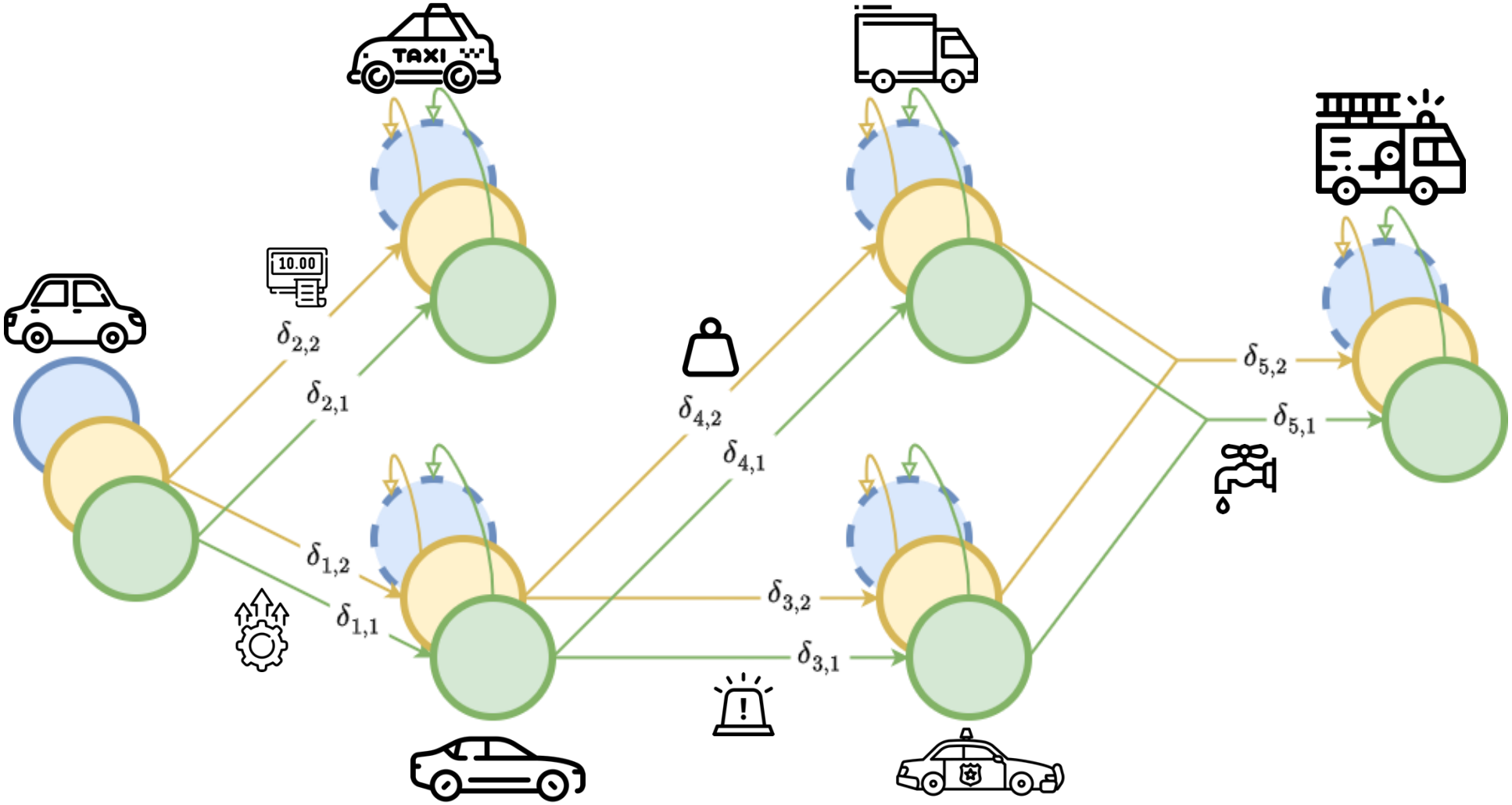
CONSISTENCY PRESERVING SPLE – VISION



CONSISTENCY PRESERVING SPLE – VISION



CONSISTENCY PRESERVING SPLE – VISION



nsistency preservation simplifies delta-oriented software product line engineering.

GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft – SFB-1608 – 501798263

CONVIDE CRC 1608

THANK YOU!

TECHNISCHE UNIVERSITÄT DRESDEN

Technische Universität München

UNIVERSITY OF MANNHEIM

KIT Karlsruher Institut für Technologie

GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft – SFB-1608 – 501798263

CONVIDE CRC 1608

RESEARCH AREA C: ENGINEERING WITH CONSISTENCY

Albert Albers (KIT-IPEK)
Bernhard Beckert (KIT-KASTEL), Tobias Düser (KIT-IPEK), Anne Koziolek (KIT-KASTEL), Ralf Reussner (KIT-KASTEL), Eric Sax (KIT-ITIV), Ina Schaefer (KIT-KASTEL)

CRC/SFB 1608
Winter Colloquium

***Defined generations** & **test bench** of braking system*

***15 Inconsistency Situations** based on Workshops*

12** Submitted, accepted or published **publications

***Taxonomy** regarding under-standings in CPS-Development*

CONVIDE
CRC 1608

C01

Engineering Consistent CPS Generations

PI

Researchers

Strategic Project C01-S

CONSISTENCY IN CROSS-GENERATIONAL ENGINEERING OF CYBER-PHYSICAL SYSTEMS

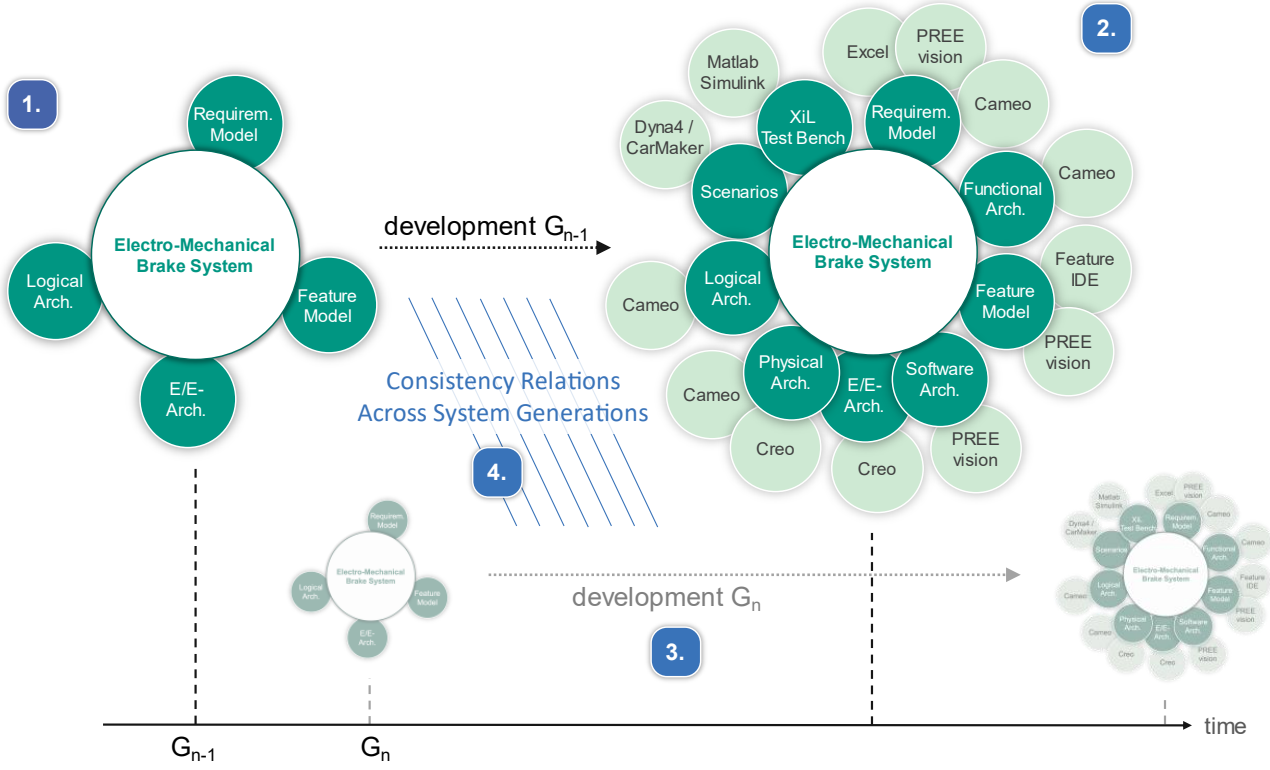
CONVIDE CRC 1608

1. Initial models for G_{n-1} **integrated into a V-SUM**

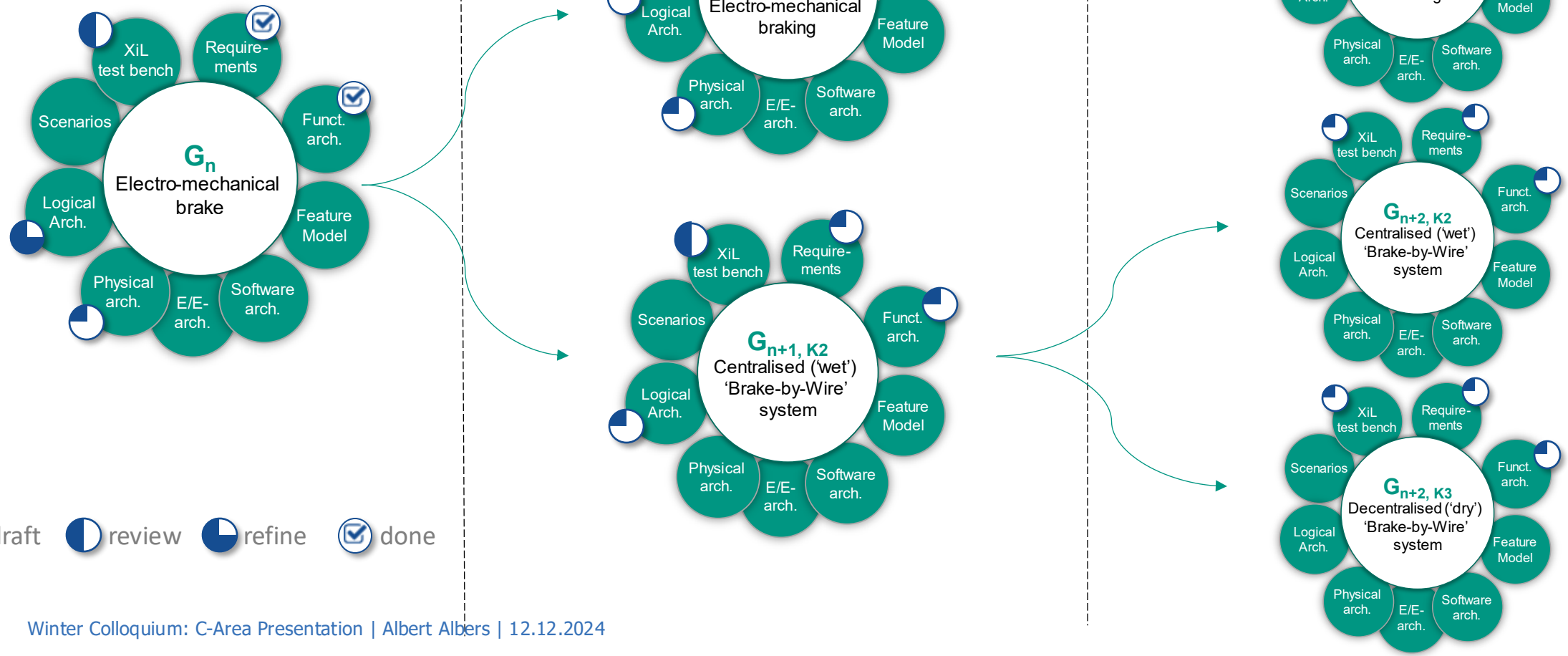
2. Further models added to complete system generation G_{n-1} to **capture key features** and **identify potential inconsistencies**

3. Create additional system generation (G_n) and analyse **possible inconsistencies between system generations**

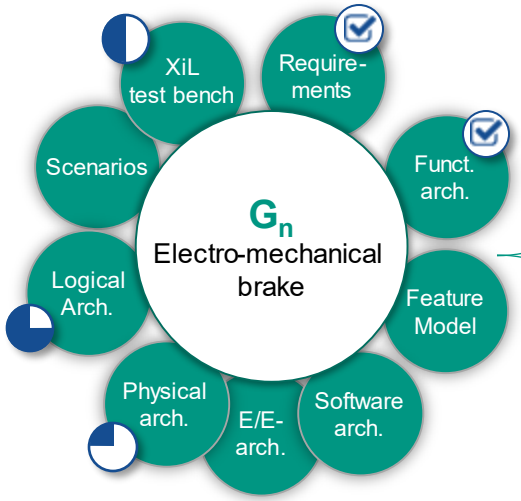
4. Identify **consistency relationships between system generations** and develop **metrics to quantify changes**

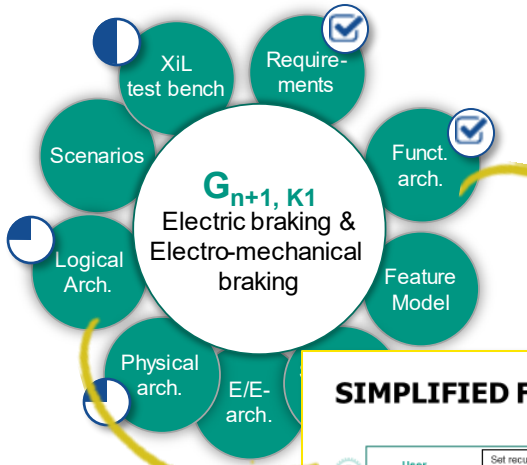


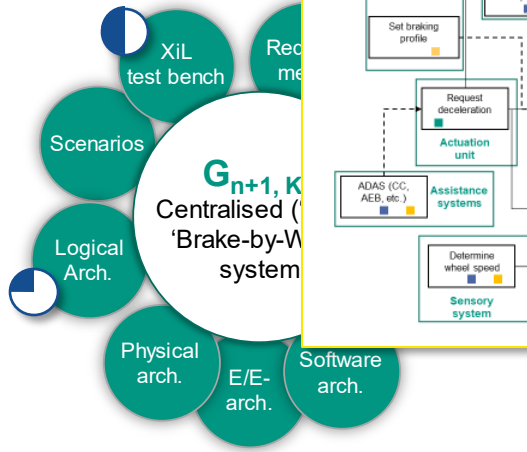
CROSS-GENERATIONAL DEVELOPMENT OF THE RESEARCH PLATFORM – DEFINED GENERATIONS

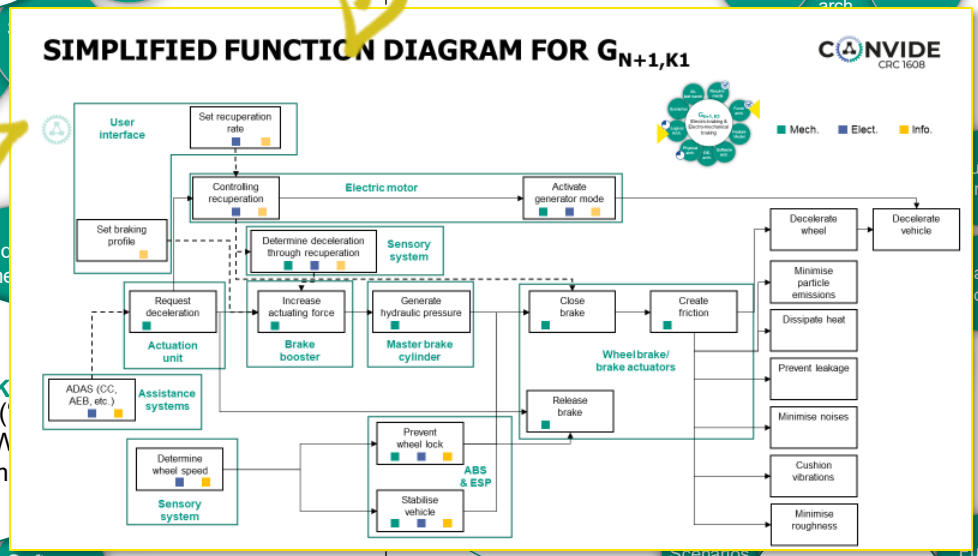


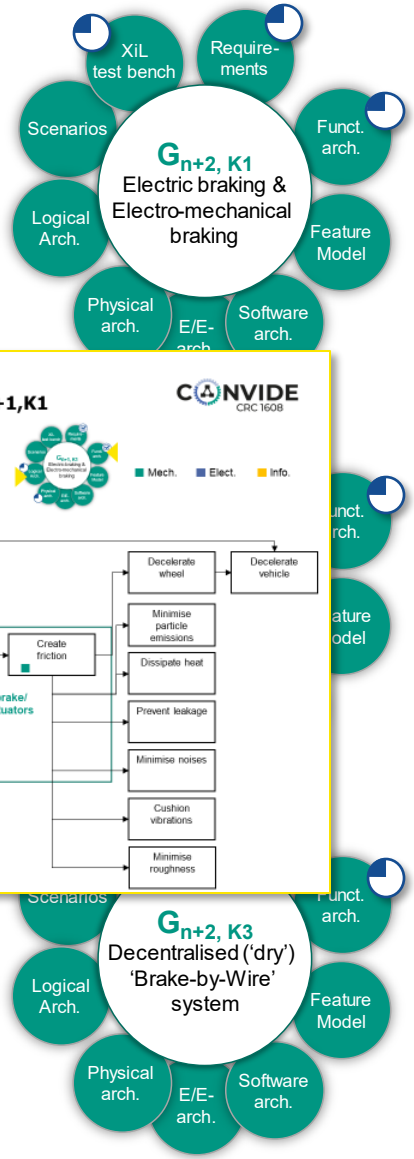
CROSS-GENERATIONAL DEVELOPMENT OF THE RESEARCH PLATFORM – CREATING VIEWS





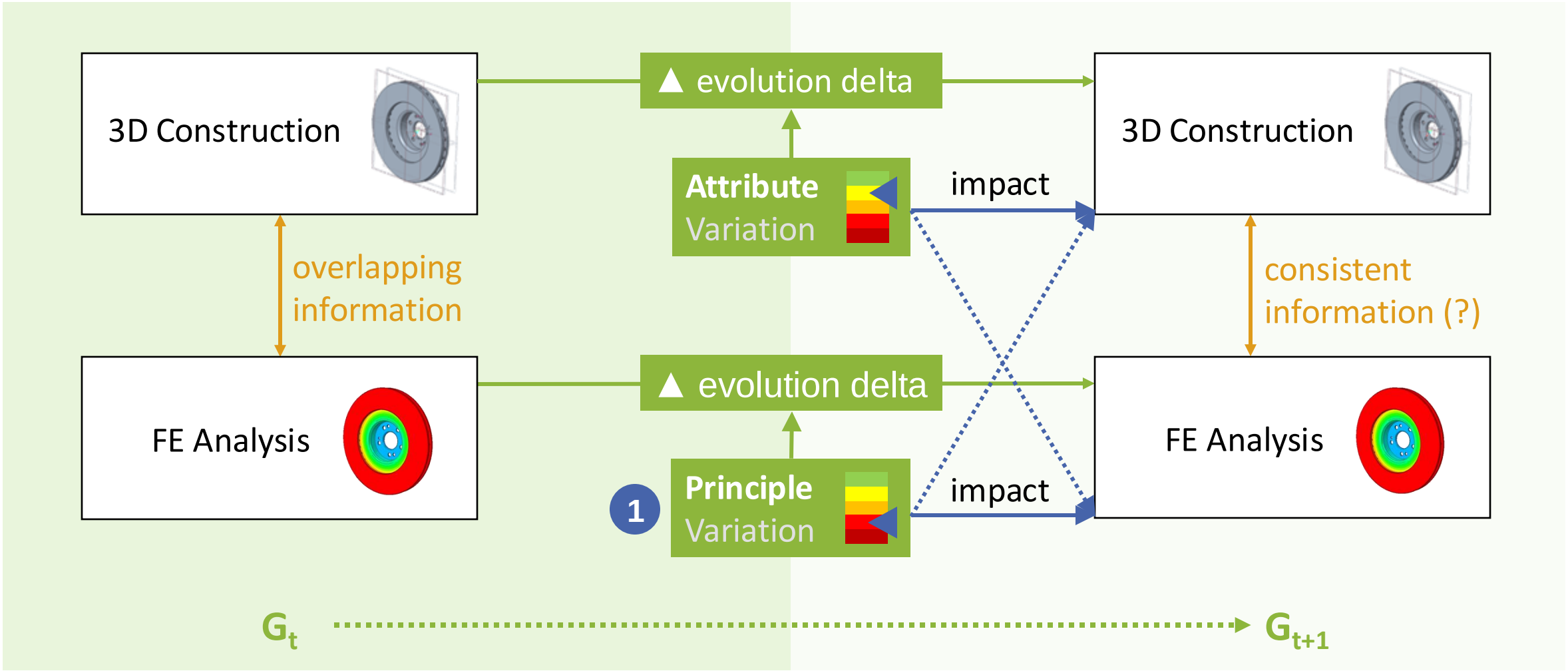






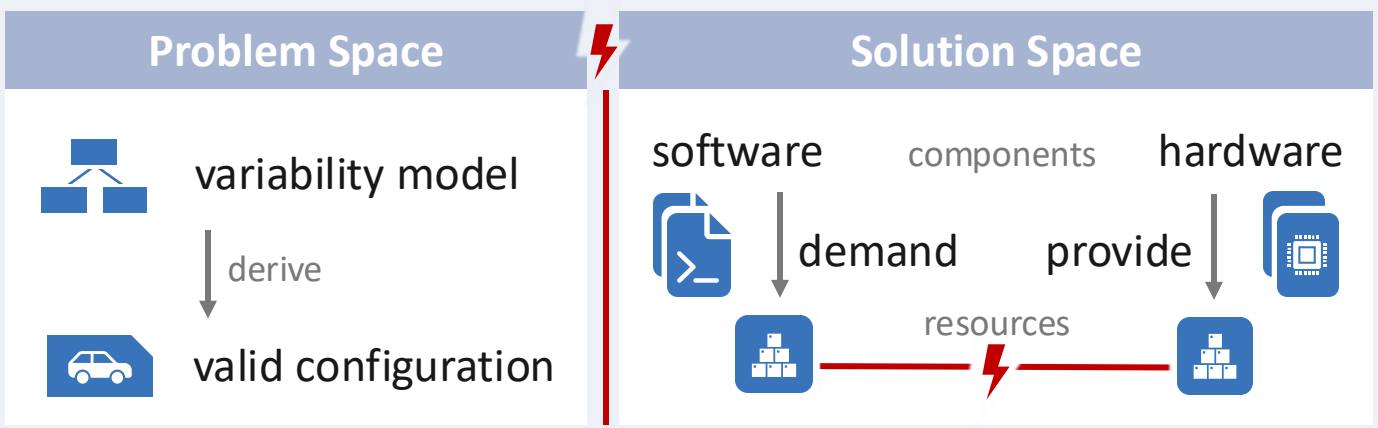
draft review refine done

APPLICATION OF DELTA-MODELLING USING MODEL OF SGE BY ALBERS



CONSISTENCY IN CYBER-PHYSICAL PRODUCT LINES

Challenge & Approach



- valid configurations can yield **incompatible artefacts** in solution space
- construct & solve **resource assignment problem** to decide realisability

Use Cases

Build-Time: Realisability & Consistency

Are all configurations offered to the customer functioning after production?

Product-in-Field: Update-Ability

Which product variants in field may receive an update so that they're still functioning?

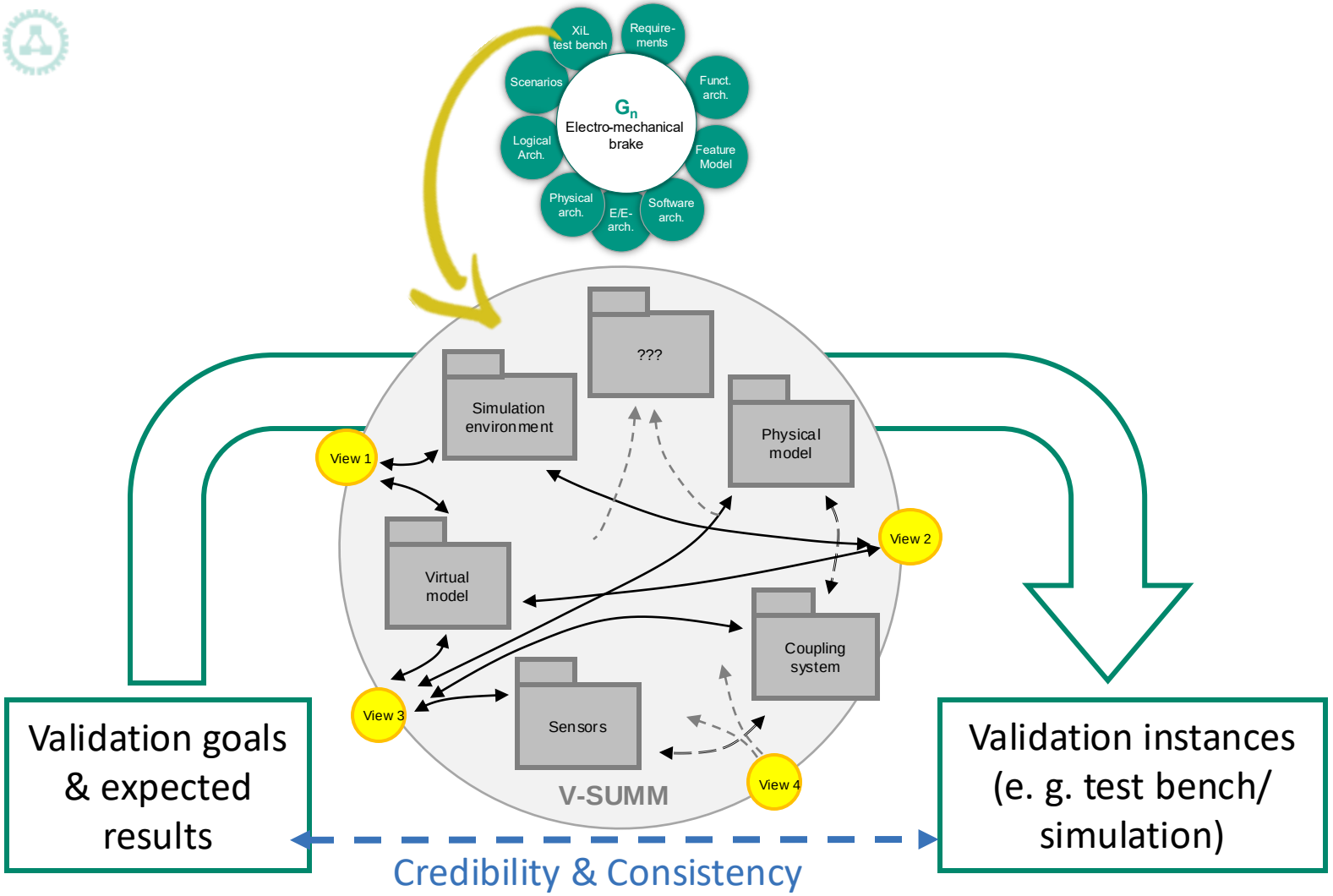
Hardware Variability

Which configurations would be non-functioning with evolved hardware?

Philip Ochs, Tobias Pett, and Ina Schaefer. 2024. Consistency Is Key: Can Your Product Line Realise What It Models?. In ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3652620.3687812.

STRATEGIC PROJECT C01-S

DEVELOPMENT OF A MODEL-BASED METHOD FOR THE INSTANTIATION OF CONSISTENT AND CREDIBLE VALIDATION ENVIRONMENTS BASED ON VALIDATION OBJECTIVES IN INTERDISCIPLINARY PRODUCT DEVELOPMENT



Steps therefore:

- Research to find existing approaches
- Modelling and using the Brake-System-in-the-Loop as lead example:
  - Reverse Engineering to find requirements for the method by going from existing validation instances back to the validation goals
  - Retrospective comparison between the existing and the generated validation instances
- Integration of the generated method in the V-SUMM approach
- Application & Evaluation on the Brake-System-in-the-Loop, KA-RaceIng and other CPS domains

CONVIDE
CRC 1608

C02

Consistency-Aware Testing of CPS Variants and Versions

PI



Researchers

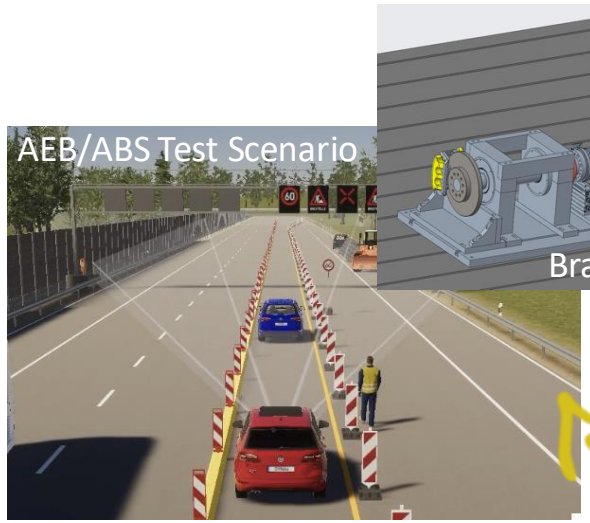


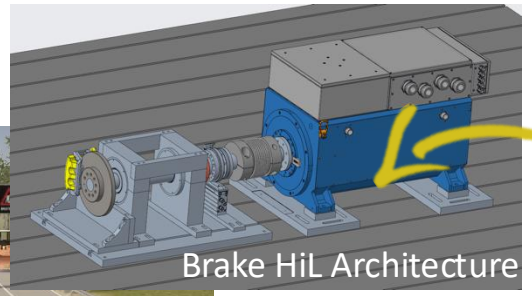


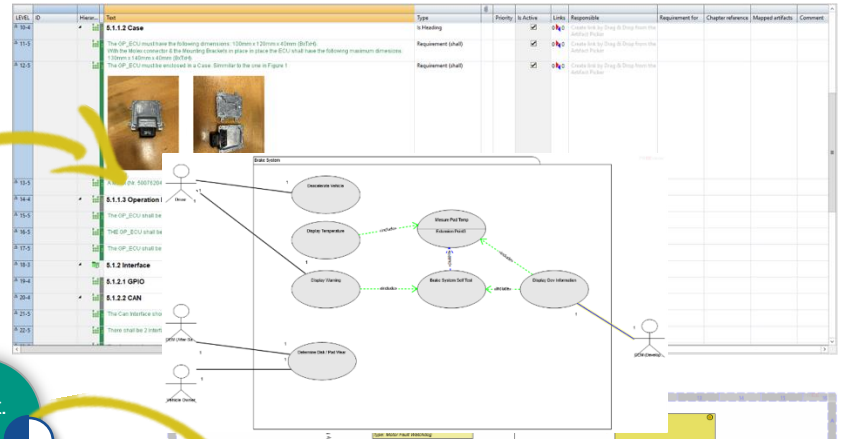


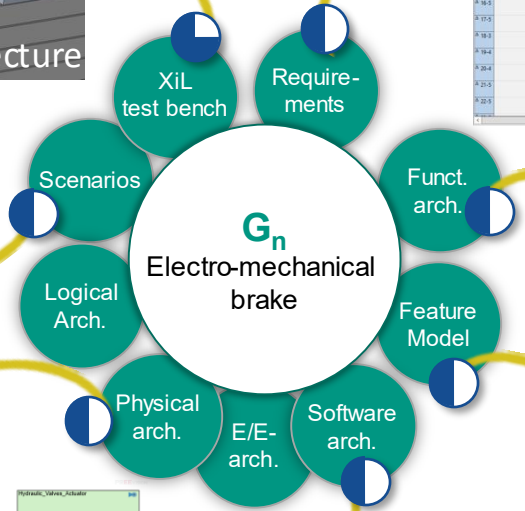
CROSS-GENERATIONAL DEVELOPMENT OF THE RESEARCH PLATFORM – VALIDATION SYSTEM

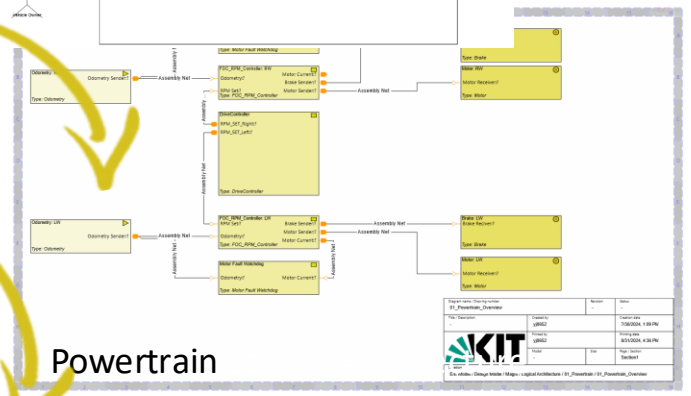
CONVIDE CRC 1608

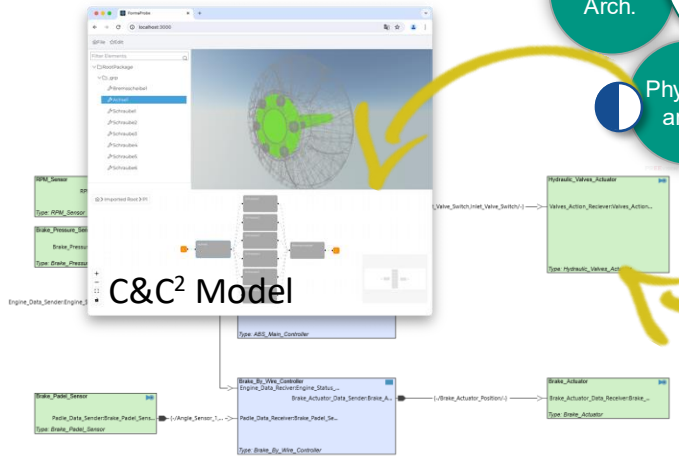


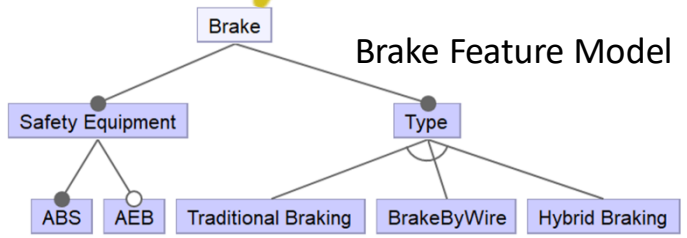


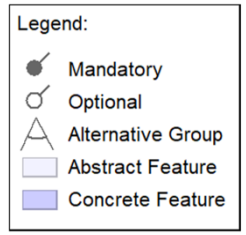








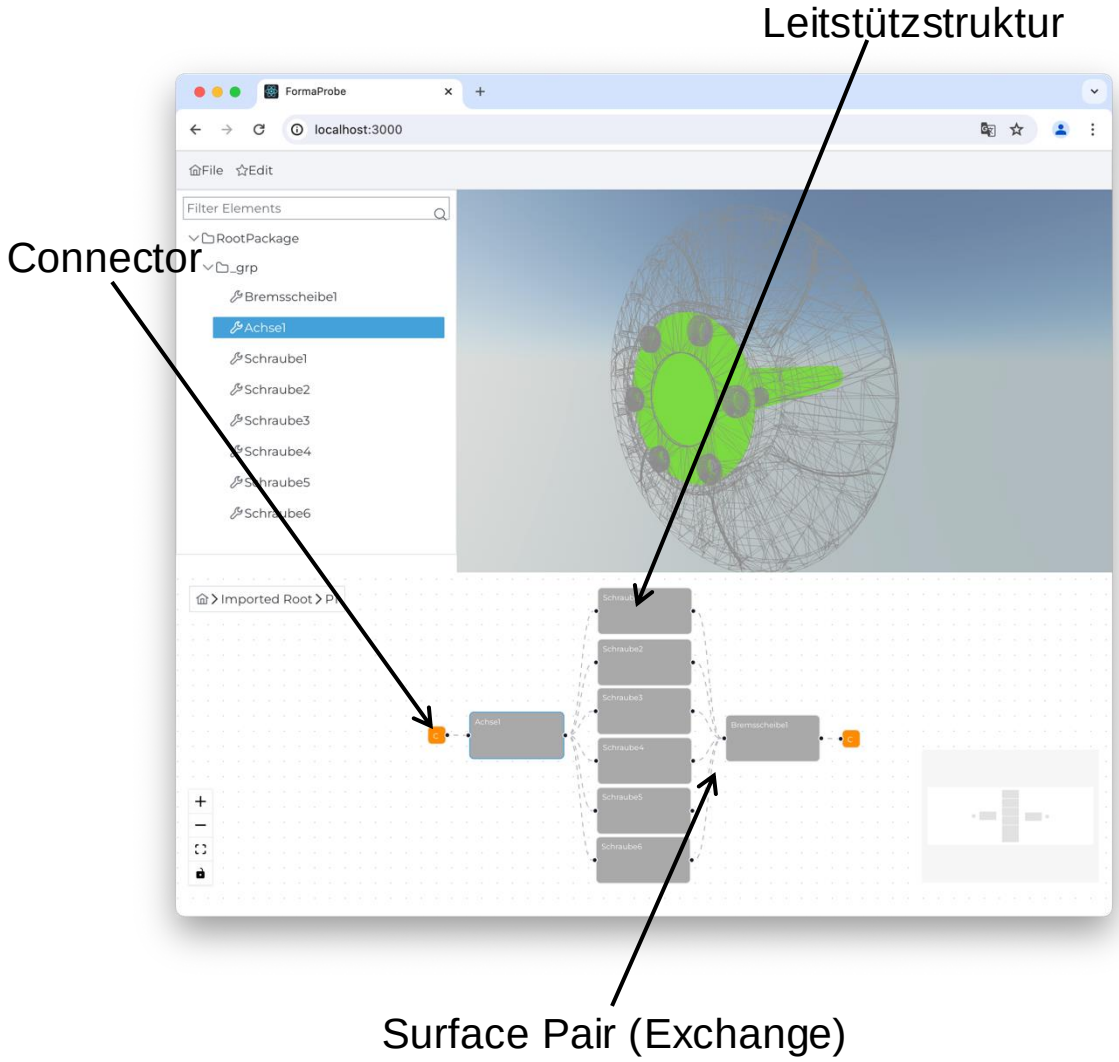




draft review refine done

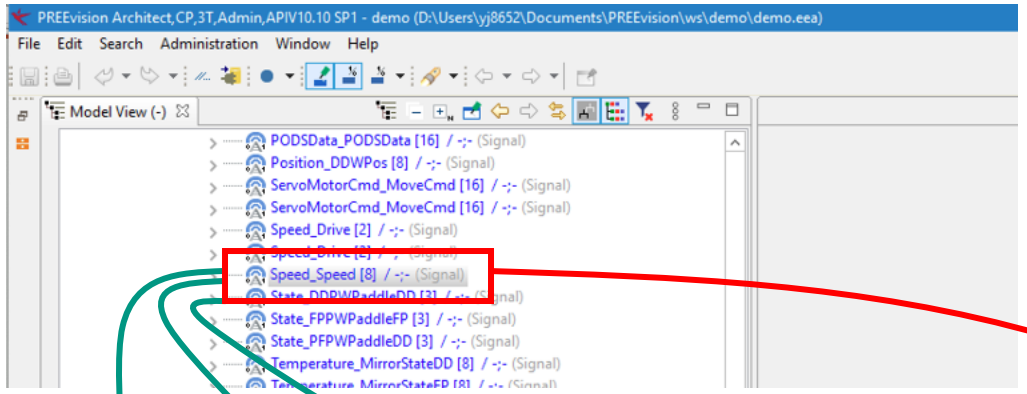
C&C² METHODOLOGY APPLIED - OVERVIEW

- Problem: CAD Models do not have a lot of Semantic Information included
- Solution: Apply C&C² Methodology to generate semantic information
- Progress:
  - Design of Ecore Meta Model for the C&C² Approach
  - Implementation of a custom C&C² View type
- Students: Master Thesis - Automatic Generation of C&C² Model from a CAD Model
- Publication @ SSP2025: Extended Abstract is submitted
- Next Steps:
  - Integration of View Type into Vitruvius
  - Integration of Models into Vitruvius
  - Definition of Consistency Relations (next slide)



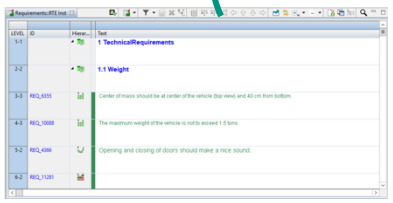
C&C² METHODOLOGY APPLIED - CONSISTENCY RELATIONS

PREEVision

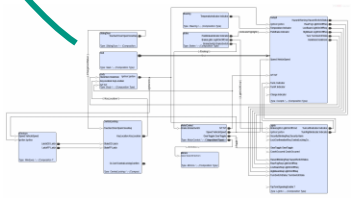


Existing Model Connections

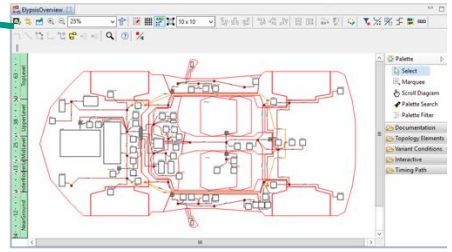
Consistency Relation



Requirements



Software

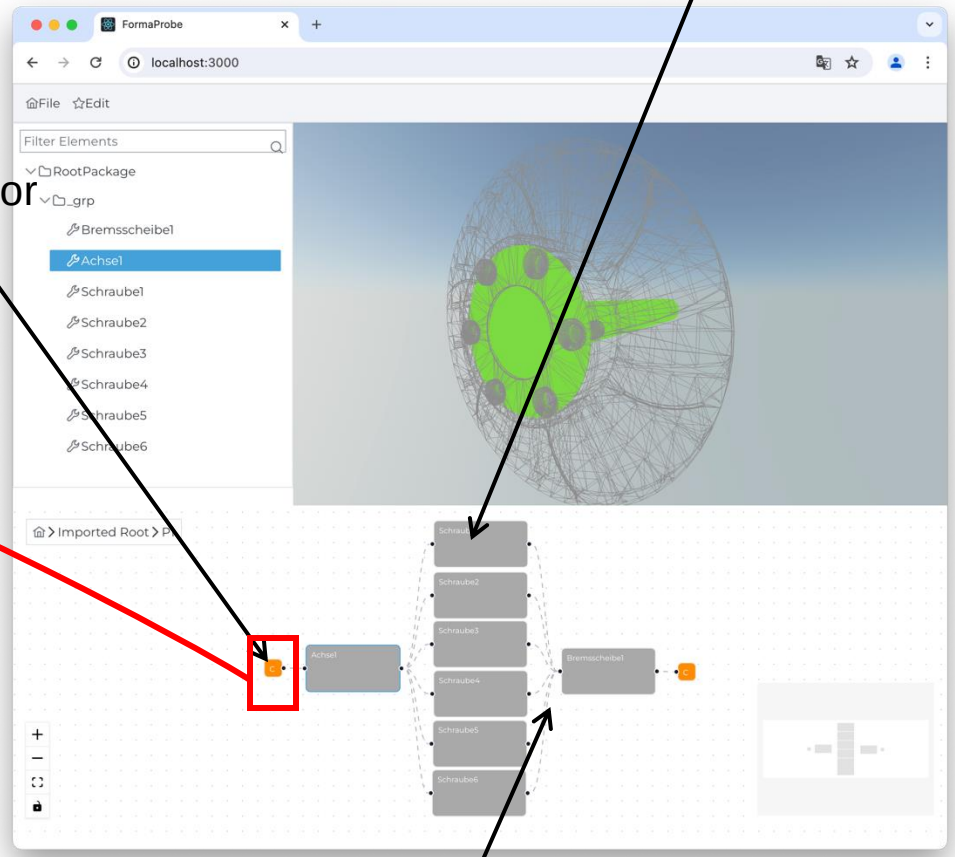


Harness / Hardware

Own C&C2 Programs

Leitstützstruktur

Connector



Surface Pair (Exchange)

CONVIDE
CRC 1608

C03

Consistency-Enabled Incremental Quality Analysis of CPS

PI



Researchers




WHY RISK ANALYSIS & ASSESSMENT MODELING LANGUAGE..?

Comprehensive Framework

- RAAML provides structured approach for safety analysis.
- Supports all modern types of safety analysis (FTA, FMEA,STPA,GSN etc.)

A Standard Language from OMG

- A proven framework eliminating the need to have customized meta models.
- Provides support to model regulatory requirements, such as ISO 26262.

Customization & Integration

- Provides extension mechanism, can be easily integrated with other models.
- Being an extension of SysML, its compatible with existing frameworks & tools.

DEVELOPMENT PROCESS OF (RAAML BASED) SAFETY ANALYSIS META-MODEL

Step-I: Preparation

- Understand RAAML
- Choose Papyrus Platform

Step II: Model Creation

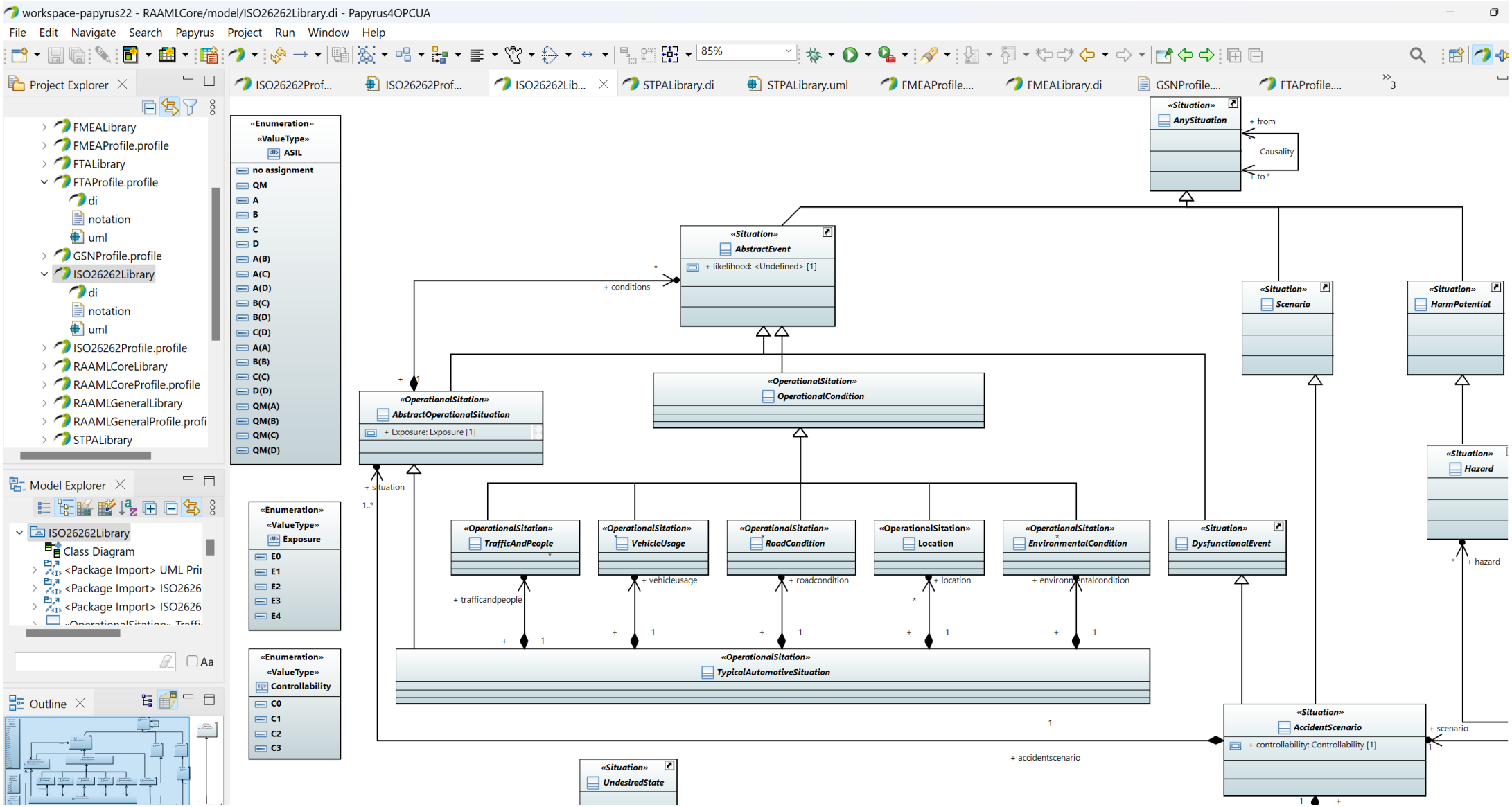
-Define stereo-types and their corresponding meta-classes.
-Use standardized notations & terminologies

Step III: Model Validation

-Model refactoring & validation

-Modularity and separation of concerns
-Version control and change management

EXAMPLE- ISO26262 LIBRARY



CONVIDE
CRC 1608

C04

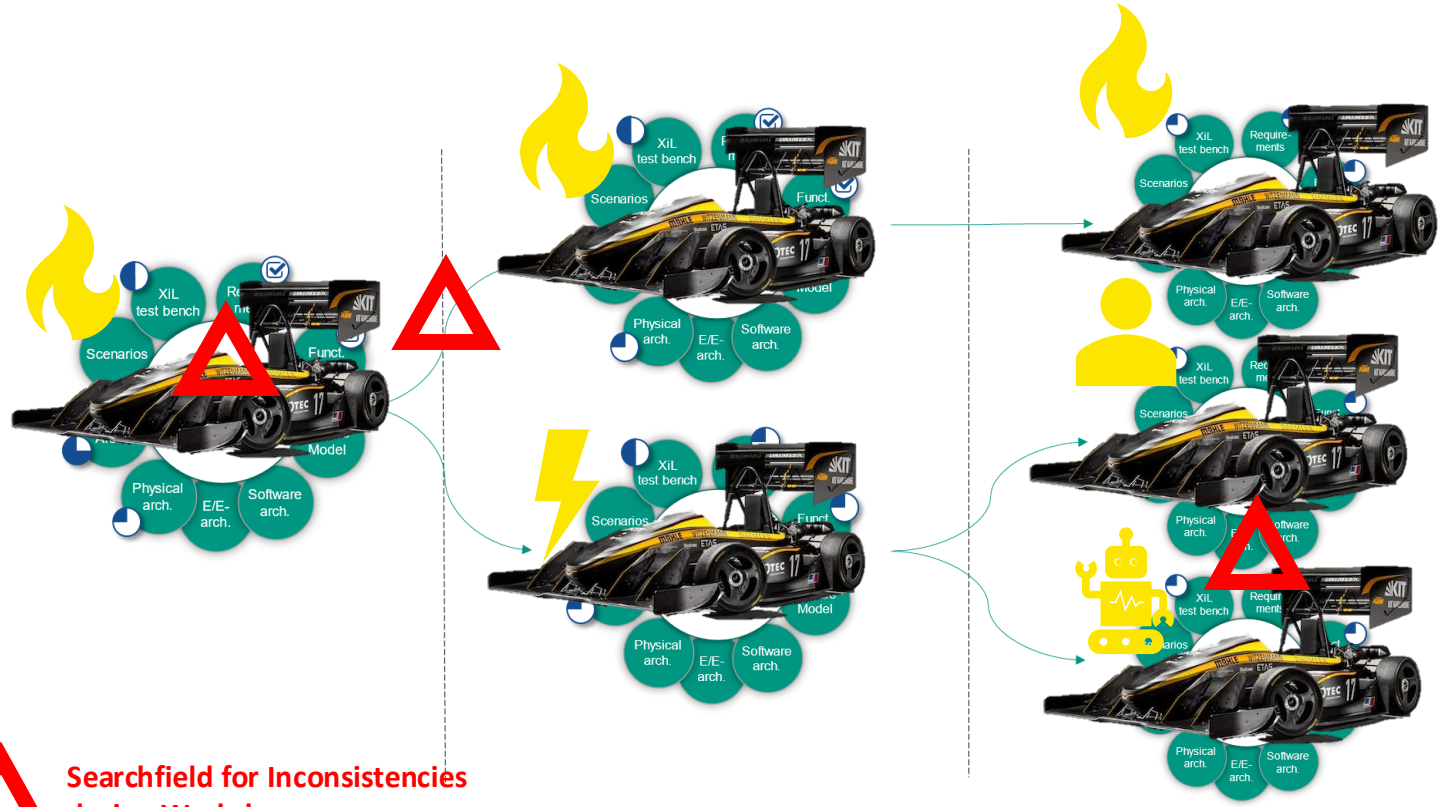
Processes for Consistent CPS Engineering

PI

Researchers

A FIRST INCONSISTENCY WORKSHOP FROM PRACTICE BRINGS A FIRST PERSPECTIVE

Workshop with 10 KA RaceIng student members



Searchfield for Inconsistencies during Workshop

No.	*Icon*	*Name of inconsistency*	*Description*
1		Uncertainty errors through the transfer of models	Inconsistency occurred in modeling (e.g., Computer-Aided Design and Computational Fluid Dynamics models), and the transferability of modeling results to reality
2		Parallel development of different models	Inconsistency occurred between the results of the different sub-teams, as individual modules were developed independently of each other
3		Independent development	Sub-teams developed subsystems independently of each other, sometimes without any knowledge of the dependencies of other subsystems.
4		Knowledge loss	Inconsistency occurred due to loss of knowledge (e.g., specific parameters in a model). Especially in a constantly changing engineering environment such as long-term student-driven development projects.
5		Unstructured communication	Inconsistency occurred due to unstructured communication between team members and information discrepancies.
6		Differences in targets and requirements documentation	Inconsistency occurred due to different target and requirements documentation between the developer and manufacturer from another branch (e.g., no data sheets available).
7		Incorrect models	Inconsistency occurred due to errors in the creation of a model. The new model is based on an incorrect or outdated model. Hence the solution does not meet the requirements and target specifications.
8		Different milestones	Inconsistency occurred due to unsynchronized development cycles and schedules (e.g., sub-team A is still in development and wants to make changes, that affect other sub-teams. Whereas, sub-team B already reached the next milestone).

Albers, A., Koziołek, A., Völk, T.A., Klippert, M., Pfaff, F., Stolpmann, R. and Schwarz, S.E., 2024, June. Identification of Inconsistencies in Agile CPS Engineering with Formula Student. In ISPIM Innovation Symposium (pp. 1-15). The International Society for Professional Innovation Management (ISPIM).

INCONSISTENCY SITUATION DESCRIPTION: TEMPLATE TO ENABLE PRACTITIONERS INPUT

Inconsistency Environment

Development Environment, Development State, Development Activities, Level of Agility, Process Models

Inconsistency Situation

Inconsistency Configuration of Models, Tools, Product Generation, Share of new development

Inconsistency Characteristics

Inconsistency Symptoms, Reoccurance, problem situation

Inconsistency Results and Learnings

Measures and impact after inconsistency observation, discovery, need for support

*There are a total of **15 interviews planned**. Currently, **9 interviews** (Automotive: 3 OEM and 1 Tier 1 supplier, Industrial: 1 OEM) have been **conducted**. If you have any other **interesting contacts** in the industry, please **share them with us** so that we can have **greater interdisciplinarity** in the interviews*

GEFÖRDERT DURCH DIE DFG Deutsche Forschungsgemeinschaft – SFB-1608 – 501798263

CONVIDE CRC 1608

THANK YOU FROM THE C AREA

CRC/SFB 1608
Winter Colloquium

!

TECHNISCHE UNIVERSITÄT DRESDEN

Technische Universität München

UNIVERSITY OF MANNHEIM

KIT Karlsruher Institut für Technologie