Karlsruhe Institute of Technology

Formal Foundations of Consistency in Model-Driven Development

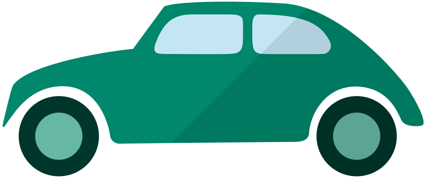
KeY Symposium 2024

R. Pascual, B. Beckert, M. Ulbrich, M. Kirsten, W. Pfeifer | July 31, 2024

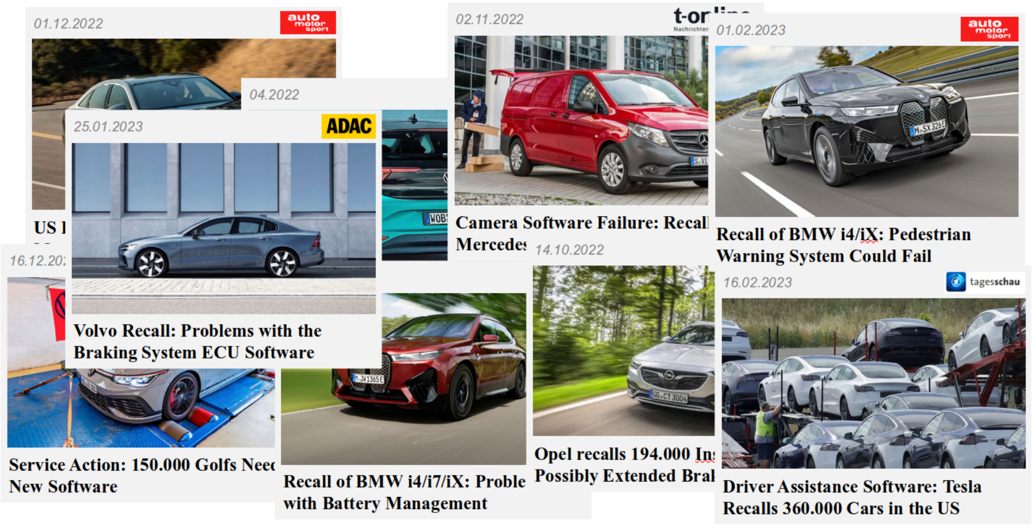
Convide[1]

Consistency in the **Vi**ew-Based **De**velopment of Cyber-Physical Systems

- Software engineering
- Mechanical engineering
- Electrical engineering
- Formal methods

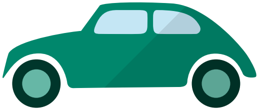


[1] Sonderforschungsbereiche (SFB) financed by the Deutsche Forschungsgemeinschaft (DFG)

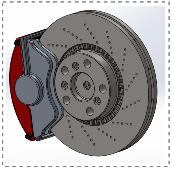

01.12.2022
auto motor sport
US I
04.2022
25.01.2023
ADAC
Volvo Recall: Problems with the Braking System ECU Software
16.12.202
Service Action: 150.000 Golfs Need New Software
Recall of BMW i4/i7/iX: Proble with Battery Management
02.11.2022
t-online
Nachrichten
Camera Software Failure: Recall Mercedes
14.10.2022
Opel recalls 194.000 In Possibly Extended Bra
01.02.2023
auto motor sport
Recall of BMW i4/iX: Pedestrian Warning System Could Fail
16.02.2023
tagesschau
Driver Assistance Software: Tesla Recalls 360.000 Cars in the US


HAVE YOU TRIED
TURNING IT OFF AND ON AGAIN?
quickmeme.com

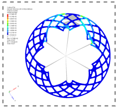
The Virtual Single Underlying Model methodology (V-SUM)



The Virtual Single Underlying Model methodology (V-SUM)

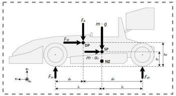


3D Brake Assembly View

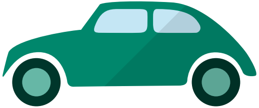


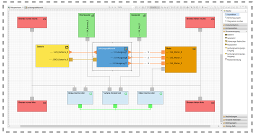
Tribological Simulation View

X-in-the-Loop Test Deployment View

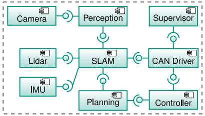


Simplified Vehicle Model View



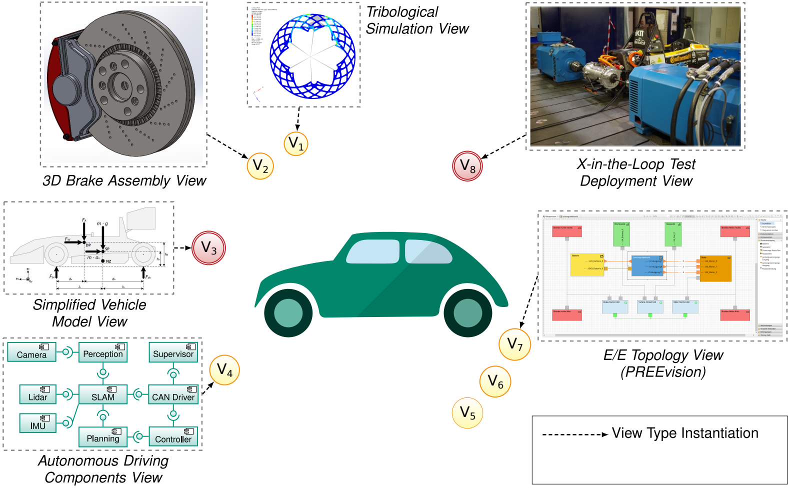


E/E Topology View (PREEvision)

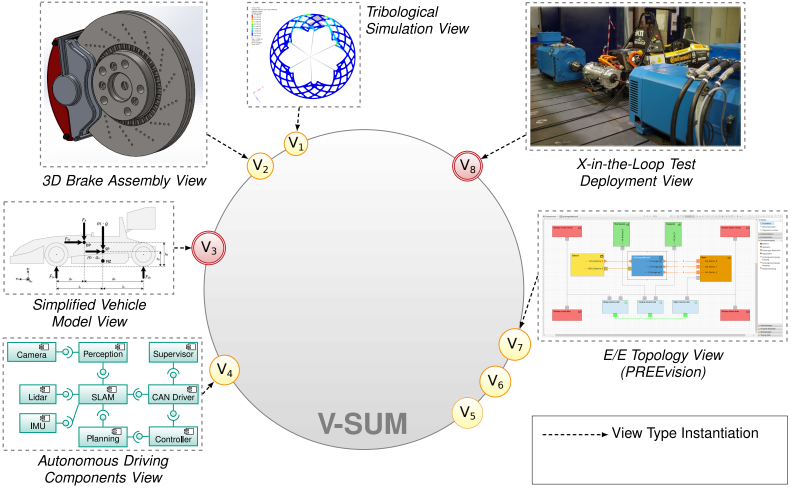


Autonomous Driving Components View

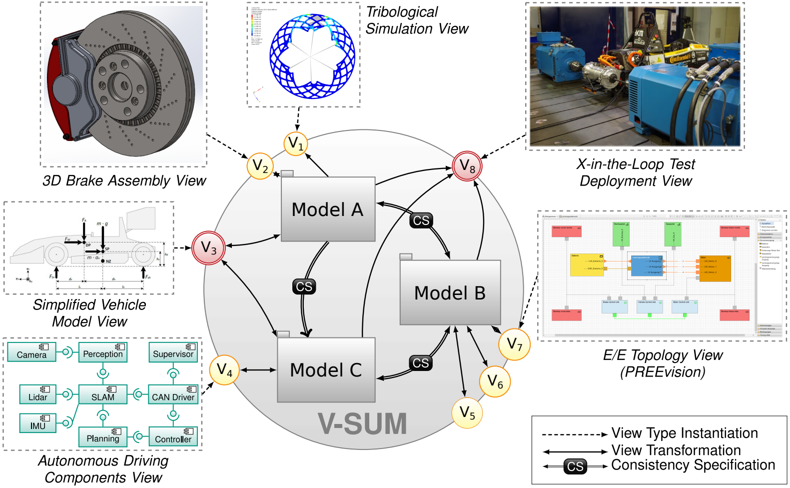
The Virtual Single Underlying Model methodology (V-SUM)



The Virtual Single Underlying Model methodology (V-SUM)



The Virtual Single Underlying Model methodology (V-SUM)



Models?

Models?

An abstract representation of an original entity (for a given purpose)

- Anything you may write with UML
- An automaton
- A Petri net
- A differential equation
- A drawing on a napkin

Models?

An abstract representation of an original entity (for a given purpose)

- Anything you may write with UML
- An automaton
- A Petri net
- A differential equation
- A drawing on a napkin

Hypothesis: models are atomic entities (we do not care about model elements)

Models?

An abstract representation of an original entity (for a given purpose)

- Anything you may write with UML
- An automaton
- A Petri net
- A differential equation
- A drawing on a napkin

Hypothesis: models are atomic entities (we do not care about model elements)

The set of syntactically admissible models is described by a **meta-model**, e.g., a formal grammar

What is even consistency?

What is even consistency?

Consistency is . . .

- From **logic**: co-satisfiability of formulae

What is even consistency?

Consistency is . . .

- From **logic**: co-satisfiability of formulae
- From **distributed development workflows**: the absence of merge conflict

What is even consistency?

Consistency is . . .

- From **logic**: co-satisfiability of formulae
- From **distributed development workflows**: the absence of merge conflict
- From **model transformations**: obtained via rule computation

What is even consistency?

Consistency is . . .

- From **logic**: co-satisfiability of formulae
- From **distributed development workflows**: the absence of merge conflict
- From **model transformations**: obtained via rule computation

Understand different notions of consistency and their properties and the induced complexity for V-SUM design, with a focus on specific aspects of CPS

What is even consistency?

Consistency is . . .

- From **logic**: co-satisfiability of formulae
- From **distributed development workflows**: the absence of merge conflict
- From **model transformations**: obtained via rule computation

Understand different notions of consistency and their properties and the induced complexity for V-SUM design, with a focus on specific aspects of CPS

We abstract **consistency** as a relation between models

Formalizing the V-SUM approach

- A set-theoretic approach to V-SUM consistency

Definitions

- A **meta-model** M_i is the set of its well-formed models $m_i \in M_i$

Definitions

- A **meta-model** M_i is the set of its well-formed models $m_i \in M_i$
- A **consistency relation** is a relation on a (finite) number of meta-models: $CR \subseteq \prod_{i \leq n} M_i$

Definitions

- A **meta-model** M_i is the set of its well-formed models $m_i \in M_i$
- A **consistency relation** is a relation on a (finite) number of meta-models: $CR \subseteq \prod_{i \leq n} M_i$
- A **V-SUM meta-model** is a pair $\mathcal{M} = (M, CR)$ where $M = \prod_{i \leq n} M_i$ and $CR \subseteq M$

Definitions

- A **meta-model** M_i is the set of its well-formed models $m_i \in M_i$
- A **consistency relation** is a relation on a (finite) number of meta-models: $CR \subseteq \prod_{i \leq n} M_i$
- A **V-SUM meta-model** is a pair $\mathcal{M} = (M, CR)$ where $M = \prod_{i \leq n} M_i$ and $CR \subseteq M$
- A **V-SUM model** m of a V-SUM meta-model $\mathcal{M}$ is a tuple $m = (m_1, \ldots, m_n)$ of models $m_i \in M_i$

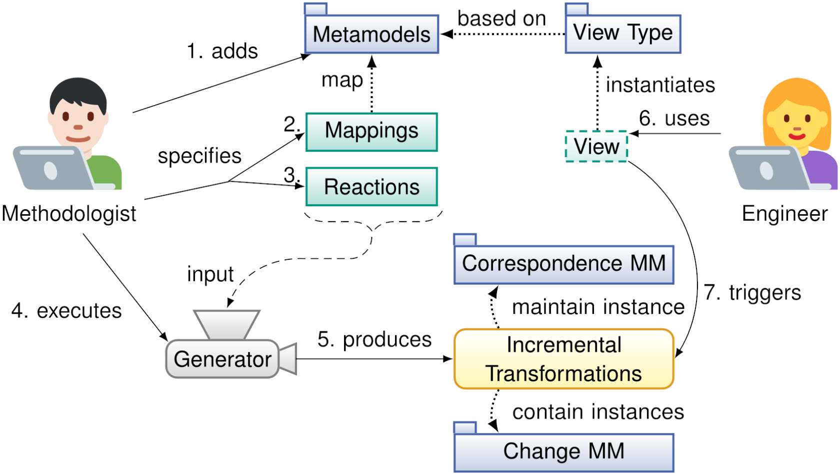
Definitions

- A **meta-model** M_i is the set of its well-formed models $m_i \in M_i$
- A **consistency relation** is a relation on a (finite) number of meta-models: $CR \subseteq \prod_{i \leq n} M_i$
- A **V-SUM meta-model** is a pair $\mathcal{M} = (M, CR)$ where $M = \prod_{i \leq n} M_i$ and $CR \subseteq M$
- A **V-SUM model** m of a V-SUM meta-model $\mathcal{M}$ is a tuple $m = (m_1, \ldots, m_n)$ of models $m_i \in M_i$
- A V-SUM model m is **consistent** wrt. CR if $m \in CR$, written $CR(m)$

How is consistency specified?

- The Vitruvius approach

Consistency Preservation with Vitruvius



Consistency from semantics

- Semantical V-SUM

An abstract notion of semantics

Examples of semantics

- **Satisfying structures** in Tarskian approach to logic,
- **Denotational** or **operational semantics** of programming languages
- **Output of a tool** as an implicit semantics for engineering models

An abstract notion of semantics

Examples of semantics

- **Satisfying structures** in Tarskian approach to logic,
- **Denotational** or **operational semantics** of programming languages
- **Output of a tool** as an implicit semantics for engineering models

Abstract semantics:

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

An abstract notion of semantics

Examples of semantics

- **Satisfying structures** in Tarskian approach to logic,
- **Denotational** or **operational semantics** of programming languages
- **Output of a tool** as an implicit semantics for engineering models

Abstract semantics:

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

- What is the codomain S?
- What is the intended meaning of $[\![\cdot]\!]$?

An abstract notion of semantics

Examples of semantics

- **Satisfying structures** in Tarskian approach to logic,
- **Denotational** or **operational semantics** of programming languages
- **Output of a tool** as an implicit semantics for engineering models

Abstract semantics:

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

- What is the codomain S?
- What is the intended meaning of $[\![\cdot]\!]$?

It is purpose-dependent, but the choice of S does not matter

How can we use semantics to define consistency?

How can we use semantics to define consistency?

Impose conditions on the semantic spaces

How can we use semantics to define consistency?

Impose conditions on the semantic spaces

Assume each meta-models M_i is equipped with an abstract semantics $[\![\cdot]\!]_i : M_i \to S_i$,
a **semantic consistency relation** is a relation $SCR \subseteq \prod_{i \leq n} S_i$

- The model $m \in M$ is semantically consistent wrt. SCR if

$$SCR([\![m_1]\!]_1, \ldots [\![m_n]\!]_n)$$

How can we use semantics to define consistency?

Impose conditions on the semantic spaces

Assume each meta-models M_i is equipped with an abstract semantics $[\![\cdot]\!]_i \colon M_i \to S_i$, a **semantic consistency relation** is a relation $SCR \subseteq \prod_{i \leq n} S_i$

- The model $m \in M$ is semantically consistent wrt. SCR if

$$SCR([\![m_1]\!]_1, \ldots [\![m_n]\!]_n)$$

We obtain a consistency relation CR_{SCR} on $\prod_{i \leq n} M_i$ (and therefore a V-SUM meta-model)

Examples

for $[\![\cdot]\!]_i$ *and* S_i

- the **set of satisfying structures** (Tarskian approach to logic)
- the **result of some tests** on a mechanical part
- the **number of methods** or **attributes** of a java class
- the **termination property** of a program

Examples

for $[\![\cdot]\!]_i$ *and* S_i

- the **set of satisfying structures** (Tarskian approach to logic)
- the **result of some tests** on a mechanical part
- the **number of methods** or **attributes** of a java class
- the **termination property** of a program

for SCR if $S_1 = S_2$

- $SCR(s_1, s_2) \iff s_1 \cap s_2 \neq \emptyset$
- $SCR(s_1, s_2) \iff s_1 \subseteq s_2$
- $SCR(s_1, s_2) \iff s_1 = s_2$

Examples

for $[\![\cdot]\!]_i$ *and* S_i

- the **set of satisfying structures** (Tarskian approach to logic)
- the **result of some tests** on a mechanical part
- the **number of methods** or **attributes** of a java class
- the **termination property** of a program

for SCR *if* $S_1 = S_2$

- $SCR(s_1, s_2) \iff s_1 \cap s_2 \neq \emptyset$
- $SCR(s_1, s_2) \iff s_1 \subseteq s_2$
- $SCR(s_1, s_2) \iff s_1 = s_2$

Allows for user-defined semantics and relations

Reasoning on semantics

- A little bit of lattice theory

Comparing semantics

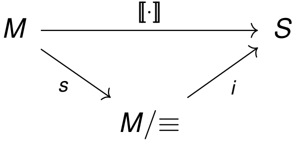
The models m_1 and m_2 are equal modulo $[\![\cdot]\!]$: $m_1 \equiv m_2 \iff [\![m_1]\!] = [\![m_2]\!]$

Comparing semantics

The models m_1 and m_2 are equal modulo $[\![\cdot]\!]$: $m_1 \equiv m_2 \iff [\![m_1]\!] = [\![m_2]\!]$
Factor out these equalities by reasoning on $M/\equiv$

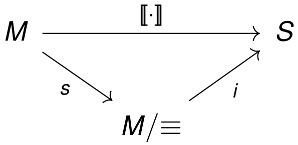
Comparing semantics

The models m_1 and m_2 are equal modulo $[\![\cdot]\!]$: $m_1 \equiv m_2 \iff [\![m_1]\!] = [\![m_2]\!]$
Factor out these equalities by reasoning on $M/\equiv$



Comparing semantics

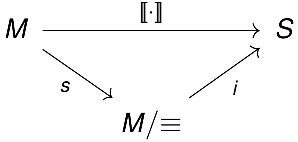
The models m_1 and m_2 are equal modulo $[\![\cdot]\!]$: $m_1 \equiv m_2 \iff [\![m_1]\!] = [\![m_2]\!]$
Factor out these equalities by reasoning on $M/\equiv$



Restricting each S_i to the image of the function $\{[\![m_i]\!]_i \mid m_i \in M_i\}$ ensures that $M_i/\equiv_i$ and S_i are isomorphic

Comparing semantics

The models m_1 and m_2 are equal modulo $[\![\cdot]\!]$: $m_1 \equiv m_2 \iff [\![m_1]\!] = [\![m_2]\!]$
Factor out these equalities by reasoning on $M/\equiv$



Restricting each S_i to the image of the function $\{[\![m_i]\!]_i \mid m_i \in M_i\}$ ensures that $M_i/\equiv_i$ and S_i are isomorphic

Reduces the study to the quotient set M/R for the equivalence relations $R \subseteq M \times M$

The example of Boolean semantics

Let $[\![\cdot]\!]_{\mathbb{B}}$ be the Boolean semantics, stating whether a model (formula) is either semantically true or false (for some logic)

The example of Boolean semantics

Let $[\![\cdot]\!]_{\mathbb{B}}$ be the Boolean semantics, stating whether a model (formula) is either semantically true or false (for some logic)

$[\![\cdot]\!]_{\mathbb{B}}$ is entirely characterized by which models get the same truth value

The example of Boolean semantics

Let $[\![\cdot]\!]_{\mathbb{B}}$ be the Boolean semantics, stating whether a model (formula) is either semantically true or false (for some logic)

$[\![\cdot]\!]_{\mathbb{B}}$ is entirely characterized by which models get the same truth value

$\mathbb{B} = \{0, 1\}$, $\{\bot, \top\}$, or $\{\bot, \top\}$ **does not matter**

The example of Boolean semantics

Let $[\![\cdot]\!]_{\mathbb{B}}$ be the Boolean semantics, stating whether a model (formula) is either semantically true or false (for some logic)

$[\![\cdot]\!]_{\mathbb{B}}$ is entirely characterized by which models get the same truth value

$\mathbb{B} = \{0, 1\}$, $\{\bot, \top\}$, or $\{\bot, \top\}$ **does not matter**

$[\![\cdot]\!]_{\mathbb{B}}$ partitions M into two equivalence classes

The example of Boolean semantics

Let $[\![\cdot]\!]_{\mathbb{B}}$ be the Boolean semantics, stating whether a model (formula) is either semantically true or false (for some logic)

$[\![\cdot]\!]_{\mathbb{B}}$ is entirely characterized by which models get the same truth value

$\mathbb{B} = \{0, 1\}$, $\{\bot, \top\}$, or $\{\bot, \top\}$ **does not matter**

$[\![\cdot]\!]_{\mathbb{B}}$ partitions *M* into two equivalence classes

For practical purposes, it **does** matter whether the binary semantics space $\mathbb{B}$ is encoded as $\{0, 1\}$, $\{\bot, \top\}$, or $\{\bot, \top\}$

The example of Boolean semantics

Let $[\![\cdot]\!]_{\mathbb{B}}$ be the Boolean semantics, stating whether a model (formula) is either semantically true or false (for some logic)

$[\![\cdot]\!]_{\mathbb{B}}$ is entirely characterized by which models get the same truth value

$\mathbb{B} = \{0, 1\}$, $\{\bot, \top\}$, or $\{\bot, \top\}$ **does not matter**

$[\![\cdot]\!]_{\mathbb{B}}$ partitions *M* into two equivalence classes

For practical purposes, it **does** matter whether the binary semantics space $\mathbb{B}$ is encoded as $\{0, 1\}$, $\{\bot, \top\}$, or $\{\bot, \top\}$

> The choice of representatives (or names) is irrelevant to compare the amount of information kept by the abstract semantics

The lattice of semantics

Theorem (Crawley and Dilworth 1973, Chap. 12 or Grätzer 2003, Sect. IV.4)

The set of all equivalence relations on a set form a complete lattice called the **equivalence lattice** with set-inclusion as order

- Meet (infimum): $\bigwedge R = \bigcap R$
- Join (supremum): $\bigvee R = (\bigcup R)^*$

The lattice of semantics

Theorem (Crawley and Dilworth 1973, Chap. 12 or Grätzer 2003, Sect. IV.4)

The set of all equivalence relations on a set form a complete lattice called the **equivalence lattice** with set-inclusion as order

- Meet (infimum): $\bigwedge R = \bigcap R$
- Join (supremum): $\bigvee R = (\bigcup R)^*$

The isomorphism transfers the lattice structure from the equivalence relations to the abstract semantics, reserving the order:

$$M/R_1 \sqsubseteq M/R_2 \iff R_2 \subseteq R_1$$

We write $\mathcal{L}_{\mathrm{sem}}^{M}$ for the lattice of semantics on M

Intuitions

Given two semantics $[\![\cdot]\!]_1$ and $[\![\cdot]\!]_2$, $[\![\cdot]\!]_1 \sqsubseteq [\![\cdot]\!]_2$ if and only if $[\![\cdot]\!]_2$ allows distinguishing between the same model as $[\![\cdot]\!]_1$ and possibly more

Intuitions

Given two semantics $[\![\cdot]\!]_1$ and $[\![\cdot]\!]_2$, $[\![\cdot]\!]_1 \sqsubseteq [\![\cdot]\!]_2$ if and only if $[\![\cdot]\!]_2$ allows distinguishing between the same model as $[\![\cdot]\!]_1$ and possibly more

The **bottom element** $[\![\cdot]\!]_\perp : M \to M/M^2 \simeq \{\star\}$ in the lattice of semantics corresponds to the trivial relation M^2 that relates any two elements

- All models have the same semantics $[\![m]\!]_\perp = \star$

Intuitions

Given two semantics $[\![\cdot]\!]_1$ and $[\![\cdot]\!]_2$, $[\![\cdot]\!]_1 \sqsubseteq [\![\cdot]\!]_2$ if and only if $[\![\cdot]\!]_2$ allows distinguishing between the same model as $[\![\cdot]\!]_1$ and possibly more

The **bottom element** $[\![\cdot]\!]_\bot : M \to M/M^2 \simeq \{\star\}$ in the lattice of semantics corresponds to the trivial relation M^2 that relates any two elements

- All models have the same semantics $[\![m]\!]_\bot = \star$

The **top element** $[\![\cdot]\!]_\top : M \to M/\mathrm{id}_M \simeq M$ corresponds to the identity relation that relates every element only to itself

- Every model $m \in M$ is its own semantic value $[\![m]\!]_\top = m$

Compatible semantics

A family of abstract semantics $(\llbracket\cdot\rrbracket_i : M_i \to S_i)_{i \leq n}$ is **compatible** with CR if and only if there is a semantic consistency relation $SCR \subseteq \prod_{i \leq n} S_i$ st.

$$CR = CR_{SCR}$$

Compatible semantics encode enough information to determine if models are consistent

Natural semantics

We consider the relation $\sim_i$ st. models are related if and only if the sets of tuples that extend them to consistent V-SUM models are the same:

$$m_a \sim_i m_b \iff CR^{\nabla i}(m_a) = CR^{\nabla i}(m_b)$$

with

$$CR^{\nabla i}(\nu) = \left\{ (m_1, \ldots, m_{i-1}, m_{i+1}, \ldots m_n) \in \prod_{j \neq i} M_j \mid CR(m_1, \ldots, m_{i-1}, \nu, m_{i+1}, \ldots m_n) \right\}$$

The semantics $[\![\cdot]\!]_i^{\mathrm{nat}} \colon M_i \to M_i/\sim_i$ are called the **natural semantics** for CR

Natural semantics

We consider the relation $\sim_i$ st. models are related if and only if the sets of tuples that extend them to consistent V-SUM models are the same:

$$m_a \sim_i m_b \iff CR^{\nabla i}(m_a) = CR^{\nabla i}(m_b)$$

with

$$CR^{\nabla i}(\nu) = \left\{ (m_1, \ldots, m_{i-1}, m_{i+1}, \ldots m_n) \in \prod_{j \neq i} M_j \mid CR(m_1, \ldots, m_{i-1}, \nu, m_{i+1}, \ldots m_n) \right\}$$

The semantics $[\![\cdot]\!]_i^{\mathrm{nat}} \colon M_i \to M_i/{\sim_i}$ are called the **natural semantics** for *CR*

Natural semantics contains just the information needed to compute *CR*

Results

Proposition 1

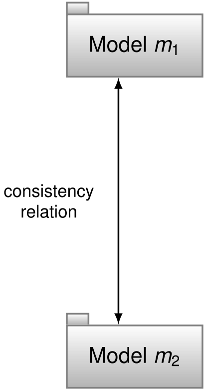
The natural semantics are compatible with *CR*

Proposition 2

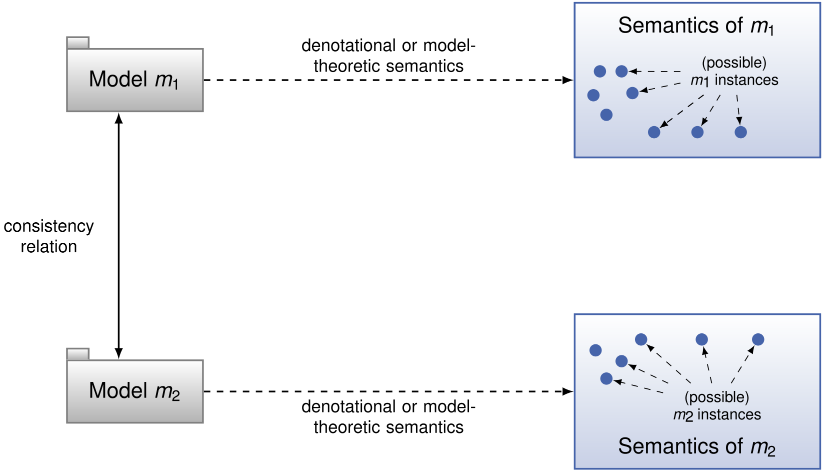
Semantics compatible with *CR* form complete lattices with the natural semantics as bottom elements

Proof: By considering $SCR^{\mathrm{nat}} = \{(\llbracket m_1 \rrbracket_1^{\mathrm{nat}}, \ldots, \llbracket m_n \rrbracket_n^{\mathrm{nat}}) \mid CR(m_1, \ldots, m_n)\}$ and the quotient sublattice (see Crawley and Dilworth 1973, Chap. 2)

Conclusion

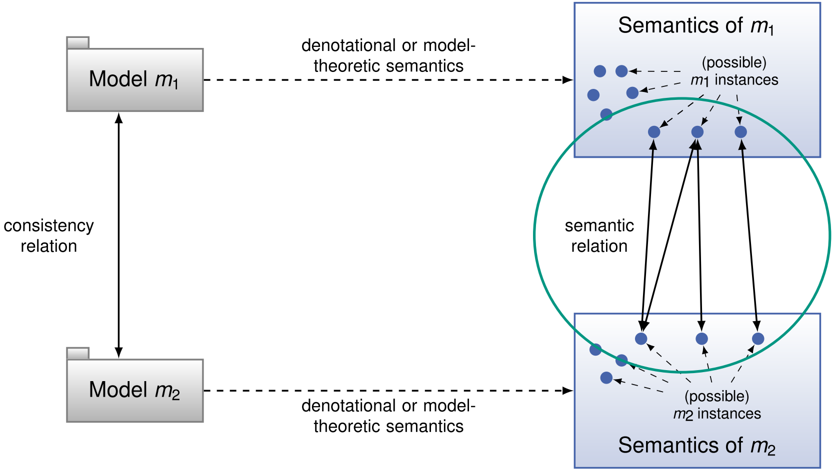


Conclusion

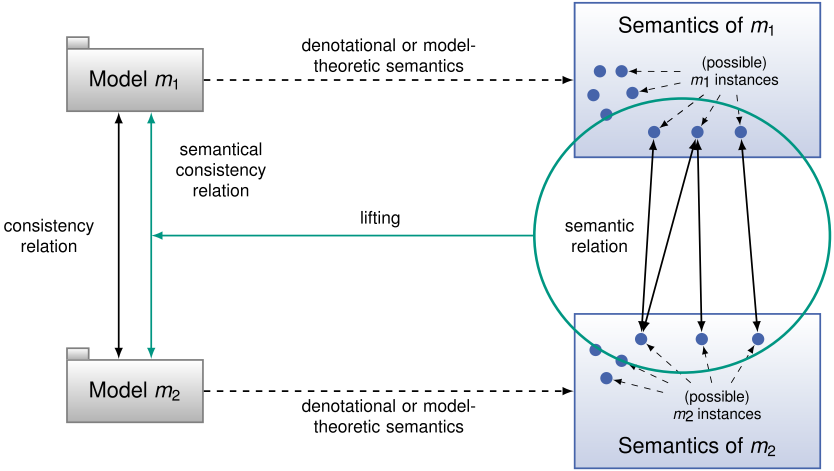


Model m_1
denotational or model-theoretic semantics
Semantics of m_1
(possible) m_1 instances
consistency relation
Model m_2
denotational or model-theoretic semantics
(possible) m_2 instances
Semantics of m_2

Conclusion

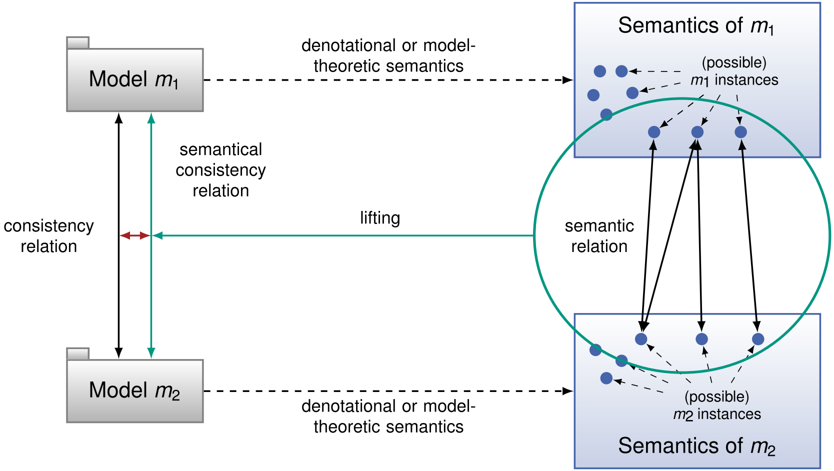

Model m_1
denotational or model-theoretic semantics
Semantics of m_1
(possible) m_1 instances
consistency relation
semantic relation
Model m_2
denotational or model-theoretic semantics
(possible) m_2 instances
Semantics of m_2

Conclusion

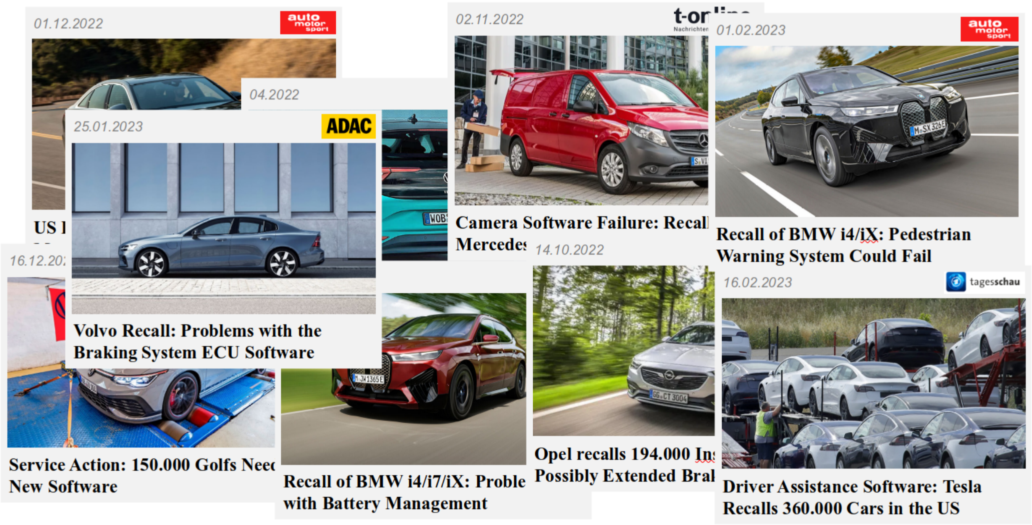


Model m_1
denotational or model-theoretic semantics
Semantics of m_1
(possible) m_1 instances
consistency relation
semantical consistency relation
lifting
semantic relation
Model m_2
denotational or model-theoretic semantics
(possible) m_2 instances
Semantics of m_2

Conclusion



Model m_1
denotational or model-theoretic semantics
Semantics of m_1
(possible)
m_1 instances
semantical consistency relation
consistency relation
lifting
semantic relation
Model m_2
denotational or model-theoretic semantics
(possible)
m_2 instances
Semantics of m_2



01.12.2022

US I

04.2022

25.01.2023

Volvo Recall: Problems with the Braking System ECU Software

16.12.202

Service Action: 150.000 Golfs Need New Software

Recall of BMW i4/i7/iX: Proble with Battery Management

02.11.2022

Camera Software Failure: Recall Mercede

14.10.2022

Opel recalls 194.000 In Possibly Extended Bral

01.02.2023

Recall of BMW i4/iX: Pedestrian Warning System Could Fail

16.02.2023

Driver Assistance Software: Tesla Recalls 360.000 Cars in the US


HAVE YOU TRIED
TURNING IT OFF AND ON AGAIN?
quickmeme.com

References I

[1] Peter Crawley and Robert P. Dilworth. *Algebraic theory of lattices*. Prentice-Hall, 1973.

[2] George Grätzer. *General Lattice Theory*. Second edition. Birkhäuser Verlag, 2003.