Configurable Formal Methods for Extreme Modeling

Uli Fahrenberg Axel Legay

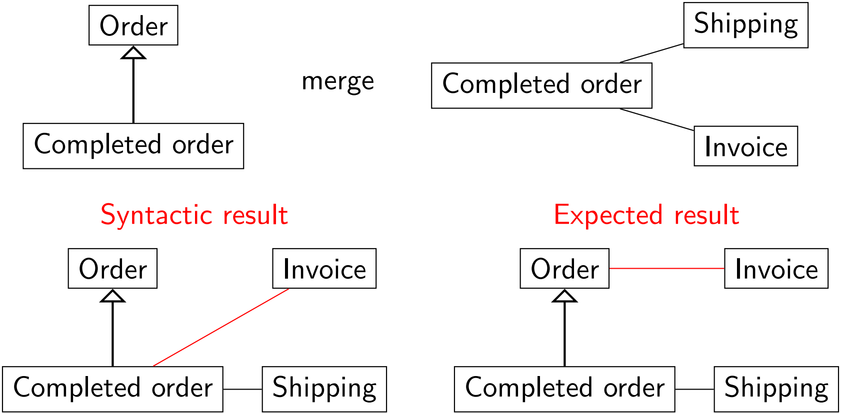
IRISA/Inria Rennes

XM 2014

Problem

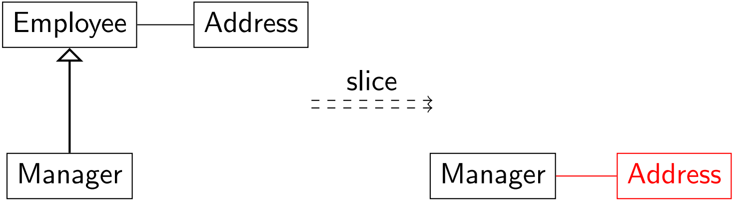
- Extreme modeling needs model transformations
- When applied uncritically, model transformations can lead to errors
- Need model transformations which are correct by design
    - or checkable by design
- But correct in relation to what semantics ??

Toy Example, Merge



How to know which associations to move up?

Toy Example, Slicing



How to know which associations to “pull down” along generalizations?

How Not to Do

Problem: Model transformations have no semantic understanding

First solution:

1. Give complete formal semantics to models
2. Define automatic and semantically correct model operators
3. PROFIT!

Yes, but:

- Complete semantics usually do not exist
- or are too complicated to be useful
- "Software engineers don't like formal methods"
- BUMMER

Proposal: bottom-up instead of top-down!

Research Proposal: My Approach

Improve existing tools for model transformations by making them partially semantics aware

- introduce only partial semantics, no more than necessary, parametrized by high-level user choices
- iterative, bottom-up approach; immediate, gradual results
- fits well with practice in industry

Application of formal methods in model-driven engineering

- but in a gentle, bottom-up way, on a need-to-know basis

Technical Example: First Steps

From our FASE 2014 paper:

- Def.: A class diagram is a tuple $\mathcal{C} = (\mathsf{cla}, \mathsf{asc}, \mathsf{gen}, \mathsf{disj}, \mathsf{ccard}, \mathsf{aends}, \mathsf{acards}) \dots$
- Def.: $\mathcal{C}_1$ refines $\mathcal{C}_2$ ($\mathcal{C}_1 \le \mathcal{C}_2$) if
  - $\mathsf{cla}_1 \supseteq \mathsf{cla}_2$, $\mathsf{asc}_1 \supseteq \mathsf{asc}_2$,
  - $\mathsf{gen}_1 \supseteq \mathsf{gen}_2$, $\mathsf{disj}_1 \supseteq \mathsf{disj}_2$,
  - $\mathsf{ccard}_1(c) \subseteq \mathsf{ccard}_2(c)$ for all $c \in \mathsf{cla}_2$,
  - $\mathsf{aends}_1(a) = \mathsf{aends}_2(a)$ for all $a \in \mathsf{asc}_2$, and
  - $\mathsf{acards}_1(a)(e) \subseteq \mathsf{acards}_2(a)(e)$ for all $a \in \mathsf{asc}_2$ and all $e \in \mathrm{dom}(\mathsf{acards}_2(a))$.

Technical Example: First Steps, contd.

(SLE 2013, contd.)

- With $\leq$, semilattice structure on class diagrams
- Greatest lower bound: merge, "$\odot$"
- Partial inverse to merge: diff, "$\setminus$"
- (Merge and diff have concrete syntactic definitions)
- Algebra similar to Girard quantale

- Using *one choice* of a globally fixed semantics,
$$\mathcal{M} \models \mathcal{C}_1 \text{ and } \mathcal{C}_1 \leq \mathcal{C}_2 \text{ imply } \mathcal{M} \models \mathcal{C}_2. \tag{1}$$
- Together with the algebraic properties above, (1) directly implies good semantic properties of merge and diff.
- Easy path to parametrization: What semantic properties are needed to prove (1)?

Variability in Semantics

How to introduce parametrized semantics:

- question of variability!
- Rumpe's idea: use feature diagram:

Semantically Configurable Consistency Analysis for CDs and ODs 159

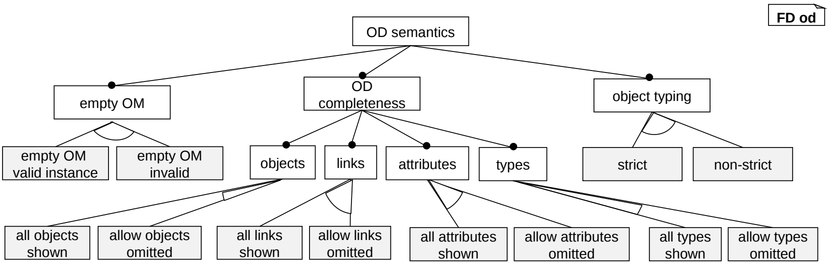


Fig. 4. The OD semantics feature diagram