Evaluating the Generality and Appropriateness of Abstractions

Dr.-Ing. Thomas Kühn

14.01.2022

DSiS - Dependebility of Software-Intensive Systems Group

Motivation
Mental Models

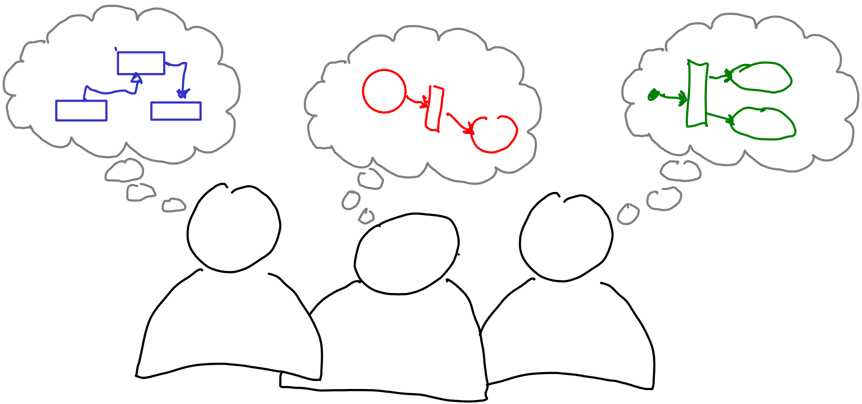


Figure 1: Different mental models formed to comprehend reality

Motivation
Common Mental Model

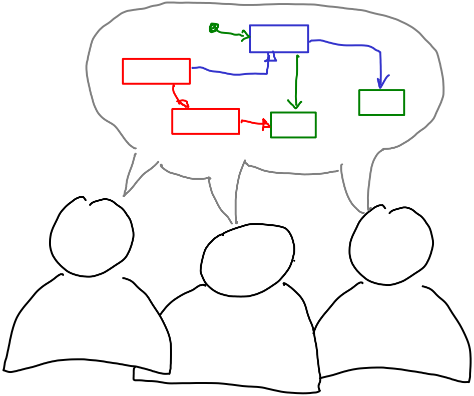


Figure 2: Discussion to come to common understanding (mental model)

Motivation

Models in Tools (Programs)

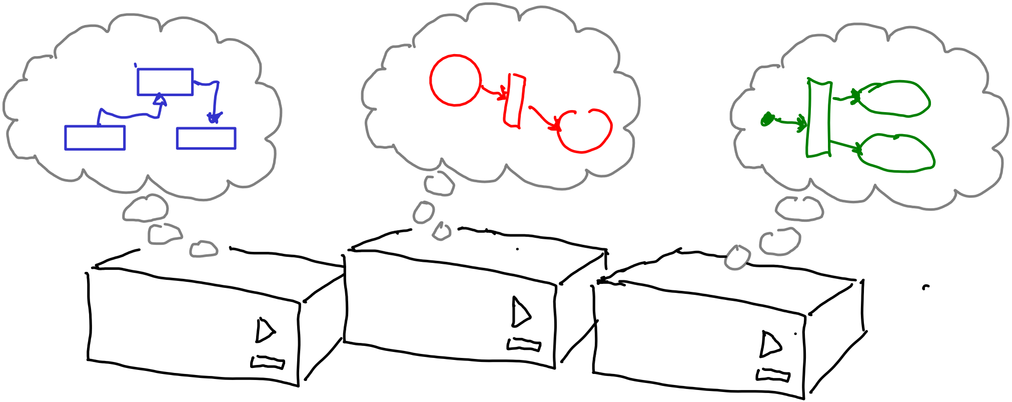


Figure 3: Mental models encoded in tools

Motivation
Abstract Model for Tools

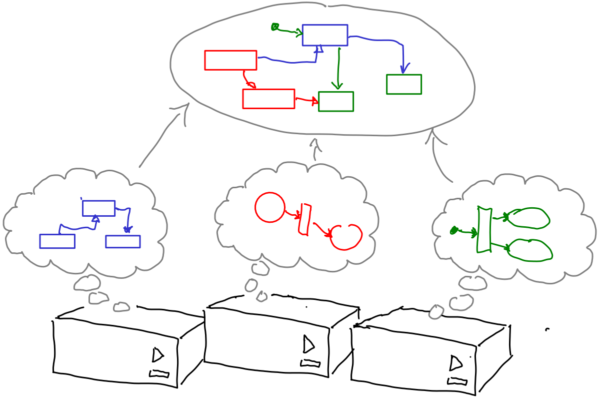


Figure 4: Unify/standardize models to an abstract model

PIBA

Karlsruhe Institute of Technology

Problem: Difficult to evaluate *generality* and *appropriateness* of abstractions

Idea: Quantify *generality* and *appropriateness* by computing fractions of abstraction covered by models in tools and vice versa

Benefit:

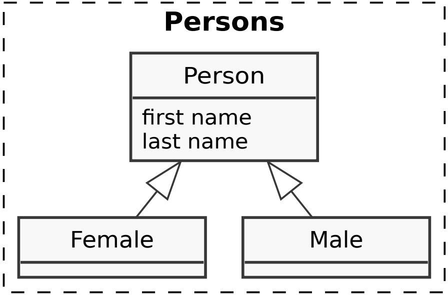
- Evaluate *generality* and *appropriateness* of abstractions
- Prevent over generalization and over specialization of abstractions

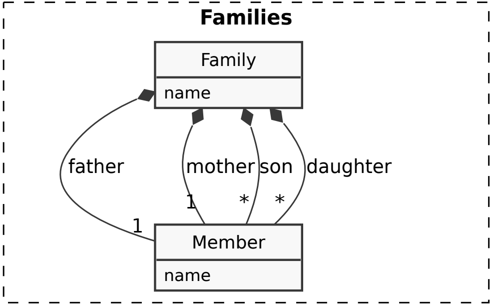
Approach:

- Metrics for abstractions based on properties by Guizzardi et al. [2005]
- Specify mapping between concepts and relations
- Compute metrics wrt. mappings

Quantifying Abstractions

Example Abstract Model and Tool Model





- Let $m \in M$ be a concept m in model M
- Let $t \in T$ be a construct t in tool T

Properties for Granularity
Laconic

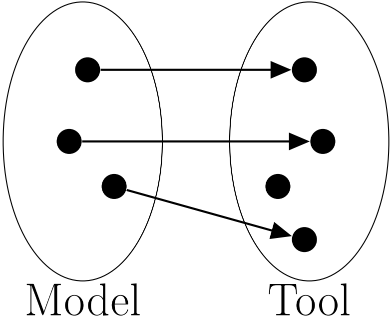


Figure 5: A tool is **laconic**, iff its constructs *t* implements at most one concept *m* of the model *M* [Guizzardi et al., 2005].

Properties for Granularity

Lucid

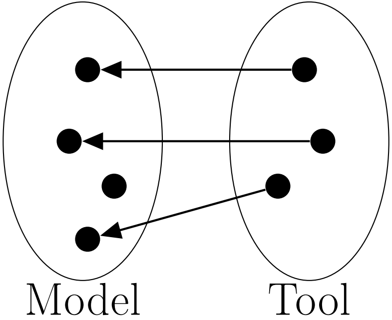


Figure 6: A model is **lucid**, iff its concepts *m* is implemented by at most one construct *t* of a tool *T* [Guizzardi et al., 2005].

Properties for Appropriateness

Complete

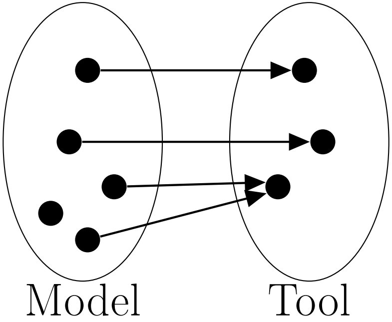


Figure 7: A tool is **complete**, iff its construct *t* is represented by at least one concept *m* in the conceptual model *M* [Guizzardi et al., 2005].

Properties for Appropriateness

Sound

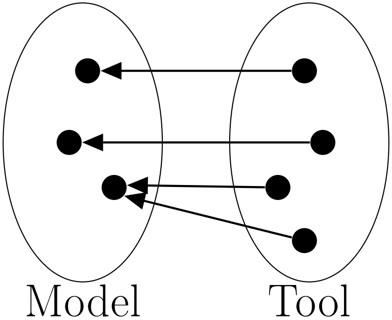


Figure 8: A model is **sound**, iff its concept m is implemented by at least one construct t in the tool T [Guizzardi et al., 2005, cf. proper].

Properties Example
Example Abstract Model and Tool Model

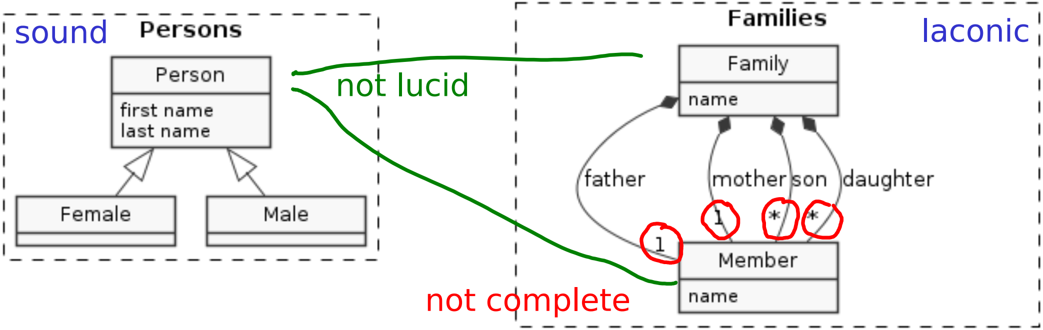


Figure 9: Abstract model is *sound* but *not lucid*; tool model is *laconic* but *not complete*.

Metrics

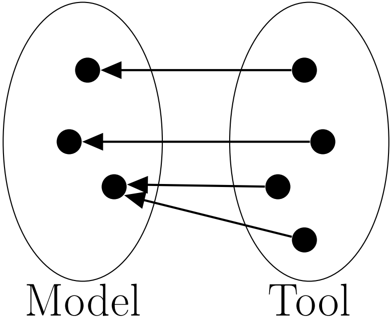


Figure 10: Mapping

- Let M be a model and T a tool
- Then $\mathbb{R}_T^M \subseteq M \times T$ is a mapping of concepts $m \in M$ to constructs $t \in T$

Metrics for Generality

Laconicity

$$\text{laconicity}(M, T) = \frac{\sum_{t \in T} \text{laconic}(M, T, t)}{|T|}$$

$$\text{laconic}(M, T, t) = \begin{cases} 1 & \textbf{if } |\{m \mid (m, t) \in \mathbb{R}_T^M\}| \leq 1 \\ 0 & \textbf{otherwise} \end{cases}$$

Lucidity

$$\text{lucidity}(M, T) = \frac{\sum_{m \in M} \text{lucid}(M, T, m)}{|M|}$$

$$\text{lucid}(M, T, m) = \begin{cases} 1 & \textbf{if } |\{t \mid (m, t) \in \mathbb{R}_T^M\}| \leq 1 \\ 0 & \textbf{otherwise} \end{cases}$$

Metrics for Appropriateness

Completeness

$$\text{completeness}(M, T) = \frac{\sum_{t \in T} \text{complete}(M, T, t)}{|T|}$$

$$\text{complete}(M, T, t) = \begin{cases} 1 & \textbf{if } |\{m \mid (m, t) \in \mathbb{R}_T^M\}| \geq 1 \\ 0 & \textbf{otherwise} \end{cases}$$

Soundness

$$\text{soundness}(M, T) = \frac{\sum_{m \in M} \text{sound}(M, T, m)}{|M|}$$

$$\text{sound}(M, T, m) = \begin{cases} 1 & \textbf{if } |\{t \mid (m, t) \in \mathbb{R}_T^M\}| \geq 1 \\ 0 & \textbf{otherwise} \end{cases}$$

Metrics Example
Example Abstract Model and Tool Model

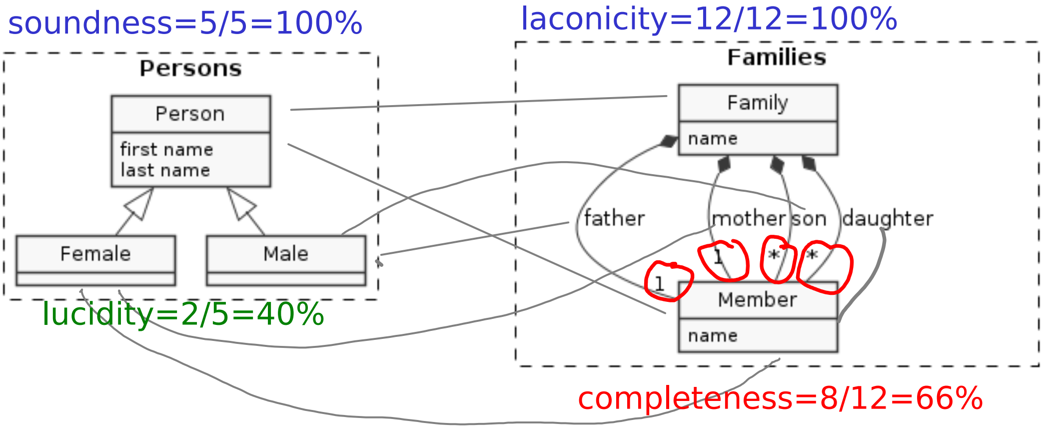


Figure 11: Abstract models *laconicity*, *lucidity*, *soundness*, *completeness*.

Application Areas

Karlsruhe Institute of Technology

Unified Model for Version and Variability Management Systems

- Establish *Unified Conceptual Model* in workshop discussions.
- Interview tool developers to create mapping to tool's model.
- Quantify *appropriateness* and *generality* of *Unified Conceptual Model* wrt. to selected tools [Ananieva et al., 2020]

Application Areas

Comparison of Taxonomies[1]

- Systematic literature review collecting taxonomies for scientific publications in software engineering
- Define *"standardized"* universal taxonomy for scientific publications in software engineering
- Create mappings between universal and identified taxonomies
- Evaluate *appropriateness* and *generality* of universal taxonomy

[1] Work in progress

Limitations

- Manually created mappings might introduce bias
- (Currently) only applicable for structural models (concepts and relations), but not for behavioral models (processes and operations)
- While suitable for vertical abstractions, limited suitability for horizontal abstractions

Conclusion

Problem: Difficult to evaluate *generality* and *appropriateness* of abstractions

Idea: Quantify *generality* and *appropriateness* by computing fractions of abstraction covered by models in tools and vice versa

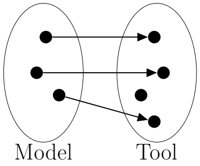
Benefit:

- Evaluate *generality* and *appropriateness* of abstractions
- Prevent over generalization and over specialization of abstractions

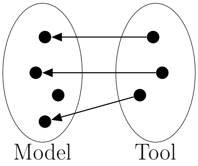
Approach:

- Establish metrics for abstractions *generality* and *appropriateness* [Ananieva et al., 2020]
- Specify mapping between concepts and relations
- Compute metrics wrt. mappings

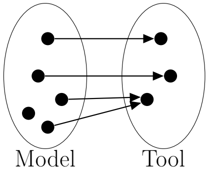
Appendix
Abstract Metric Examples



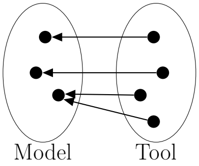
(a) Laconicity $= 1.0$



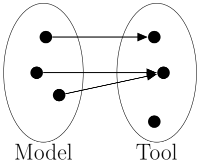
(b) Lucidity $= 1.0$



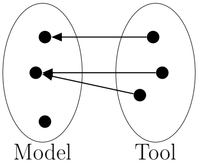
(c) Completeness $= 1.0$



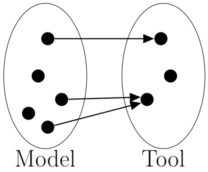
(d) Soundness $= 1.0$



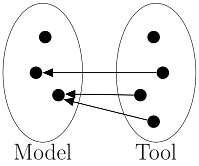
(e) Laconicity $= \frac{2}{3}$



(f) Lucidity $= \frac{2}{3}$



(g) Completeness $= \frac{2}{3}$



(h) Soundness $= \frac{2}{3}$

Generalized Metrics

Given a finite set of tools $T \in \mathcal{T}$, we can generalize these metrics, as follows:

$$\overline{\text{laconicity}}(M, \mathcal{T}) = \frac{\sum_{T \in \mathcal{T}} \sum_{t \in T} \text{laconic}(M, T, t)}{\sum_{T \in \mathcal{T}} |T|}$$

$$\overline{\text{lucidity}}(M, \mathcal{T}) = \frac{\sum_{m \in m} \left(\min_{T \in \mathcal{T}} \text{lucid}(M, T, m) \right)}{|M|}$$

$$\overline{\text{completeness}}(M, \mathcal{T}) = \frac{\sum_{T \in \mathcal{T}} \sum_{t \in T} \text{complete}(M, T, t)}{\sum_{T \in \mathcal{T}} |T|}$$

$$\overline{\text{soundness}}(M, \mathcal{T}) = \frac{\sum_{m \in M} \left(\max_{T \in \mathcal{T}} \text{sound}(M, T, m) \right)}{|M|}$$

References I

Sofia Ananieva, Sandra Greiner, Thomas Kühn, Jacob Krüger, Lukas Linsbauer, Sten Grüner, Timo Kehrer, Heiko Klare, Anne Koziolek, Henrik Lönn, Sebastian Krieter, Christoph Seidl, S. Ramesh, Ralf Reussner, and Bernhard Westfechtel. A conceptual model for unifying variability in space and time. In *Proceedings of the 24th ACM Conference on Systems and Software Product Line - Volume A*, SPLC '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375696. doi: 10.1145/3382025.3414955.

Giancarlo Guizzardi, Luís Ferreira Pires, and Marten van Sinderen. An ontology-based approach for evaluating the domain appropriateness and comprehensibility appropriateness of modeling languages. In *International Conference on Model Driven Engineering Languages and Systems*, MODELS. Springer, 2005. doi: 10.1007/11557432_51.