Formal Approach to Integrating Feature and Architecture Models

Mikoláš Janota Goetz Botterweck

Lero
University College Dublin
University of Limerick
Ireland

FASE '08

IST-15905

Introduction

Software Product Lines

- systematic development of families of similar systems
- *explicit* modeling of the family

Introduction

Software Product Lines

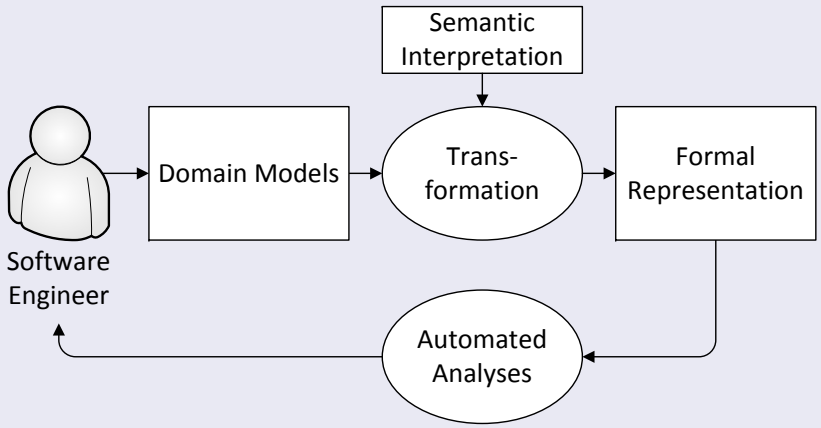
- systematic development of families of similar systems
- *explicit* modeling of the family

Modeling

- feature models — customer-oriented models
- architecture models — implementation-oriented models
- our work provides formal foundation for integrating the two

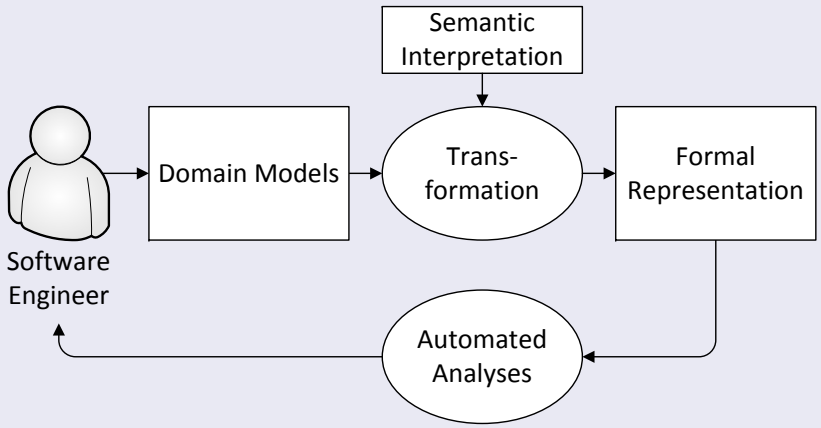
Why Formalize?

Formalisms applied in domain engineering



Why Formalize?

Formalisms applied in domain engineering

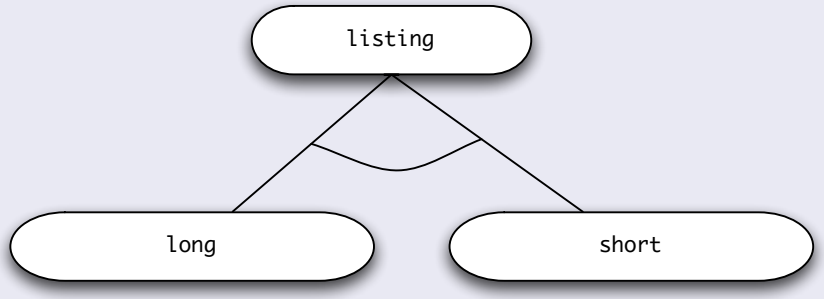


Formalisms in research

- better understanding of the relevant concepts
- relating different approaches to one another

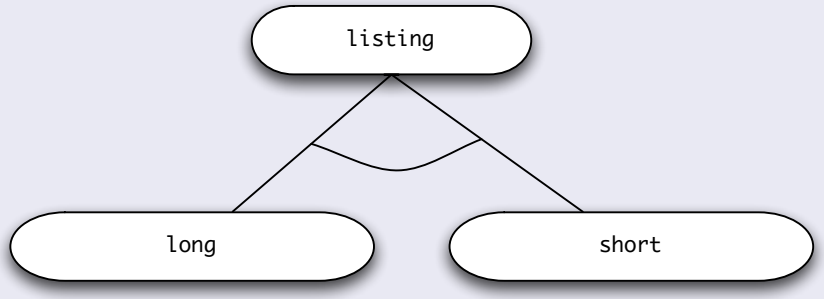
Feature Models and their Semantics

Example



Feature Models and their Semantics

Example

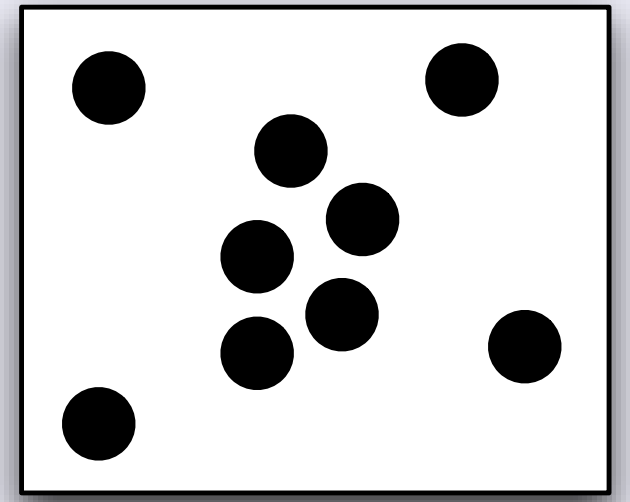


- semantics as allowed configurations

$$\emptyset, \{listing, long\}, \{listing, short\}$$

Generalization

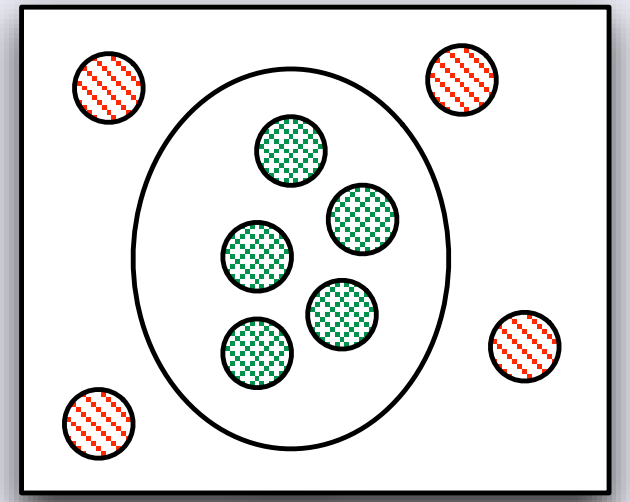
Domain



Problem space

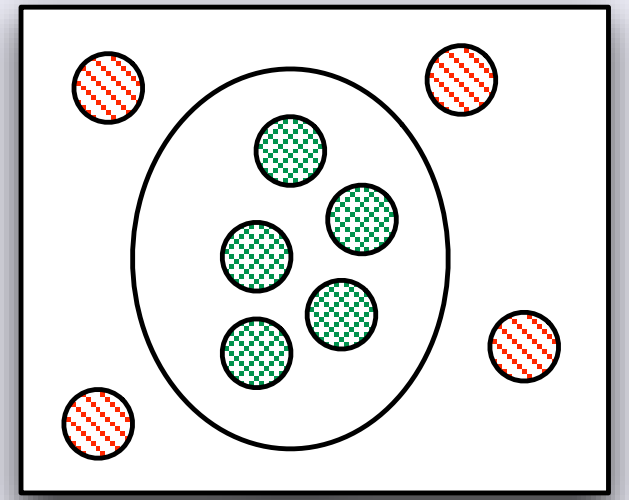
Generalization

Domain model

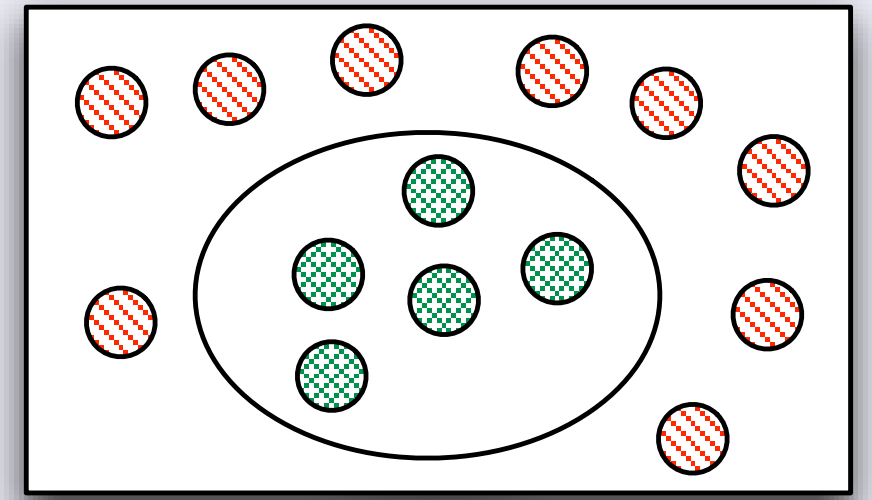


Generalization

Domain and Solution model



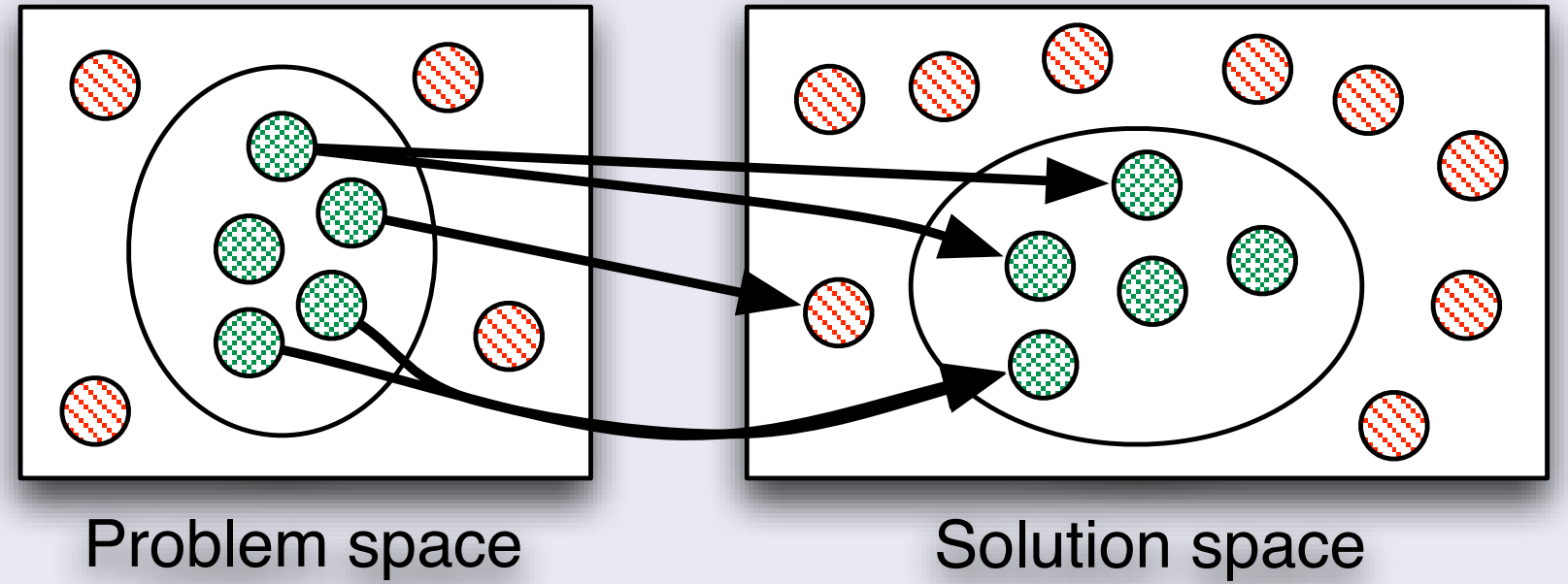
Problem space



Solution space

Generalization

Domain and Solution model combined



Models à la Math

Semantics as sets

- *feature models* are sets of investigated problems

Examples

- $\{ \{f_1\}, \{f_2\}, \{f_1, f_2\} \}$

Models à la Math

Semantics as sets

- *feature models* are sets of investigated problems
- *component models* are sets of considered solutions

Examples

- $\{ \{f_1\}, \{f_2\}, \{f_1, f_2\} \}$
- $\{ \emptyset, \{c_1\}, \{c_1, c_2\} \}$

Models à la Math

Semantics as sets

- *feature models* are sets of investigated problems
- *component models* are sets of considered solutions
- *feature-component models* are sets of pairs problem-solution

Examples

- $\{ \{f_1\}, \{f_2\}, \{f_1, f_2\} \}$
- $\{ \emptyset, \{c_1\}, \{c_1, c_2\} \}$
- $\{ \langle \{f_1\}, \{c_1\} \rangle, \langle \{f_1, f_2\}, \{c_1, c_2\} \rangle \}$

Applications

- Formalization enables precisely expressing the properties that we wish to study.

Examples

- implementable feature configurations:

$$\mathcal{I} \equiv \{\mathbf{f} \mid (\exists \mathbf{c})(\langle \mathbf{f}, \mathbf{c} \rangle \in \mathcal{M}_{\text{fc}})\}$$

Applications

- Formalization enables precisely expressing the properties that we wish to study.

Examples

- implementable feature configurations:

$$\mathcal{I} \equiv \{\mathbf{f} \mid (\exists \mathbf{c})(\langle \mathbf{f}, \mathbf{c} \rangle \in \mathcal{M}_{\mathrm{fc}})\}$$

- Is a feature model OK?

$$\mathcal{F} \subseteq \mathcal{I}$$

Defining Feature-Component Models

Split in user-friendly components

- restriction on features (problems)
- restriction on components (solutions)
- mapping between the two (realized-by)

Defining Feature-Component Models

Split in user-friendly components

- restriction on features (problems)
- restriction on components (solutions)
- mapping between the two (realized-by)

Example

$$f_1 \vee f_2 \wedge$$
$$c_2 \Rightarrow c_1 \wedge$$
$$f_1 \text{ realized-by } c_1 \wedge$$
$$f_2 \text{ realized-by } c_2$$

Realized-by

Possible interpretations

- f_1 realized-by c_1
  1. $f_1 \Rightarrow c_1$

Realized-by

Possible interpretations

- f_1 realized-by c_1
  1. $f_1 \Rightarrow c_1$
  2. $f_1 \Leftrightarrow c_1$

Realized-by

Possible interpretations

- f_1 realized-by c_1
  1. $f_1 \Rightarrow c_1$
  2. $f_1 \Leftrightarrow c_1$
  3. $f_1 \Rightarrow c_1$ but remove "unreasonable combinations"

Realized-by

Possible interpretations

- f_1 realized-by c_1
  1. $f_1 \Rightarrow c_1$
  2. $f_1 \Leftrightarrow c_1$
  3. $f_1 \Rightarrow c_1$ but remove "unreasonable combinations"

Unreasonable combinations

- with no unnecessary components or features that are not selected but are implemented
- implication interpretation
  $\langle \{f_1\}, \{c_1\} \rangle, \langle \{f_1, f_2\}, \{c_1, c_2\} \rangle,$
  $\langle \{f_2\}, \{c_1, c_2\} \rangle, \langle \{f_1\}, \{c_1, c_2\} \rangle$

Realized-by

Possible interpretations

- f_1 realized-by c_1
  1. $f_1 \Rightarrow c_1$
  2. $f_1 \Leftrightarrow c_1$
  3. $f_1 \Rightarrow c_1$ but remove "unreasonable combinations"

Unreasonable combinations

- with no unnecessary components or features that are not selected but are implemented
- implication interpretation
  $\langle \{f_1\}, \{c_1\} \rangle, \langle \{f_1, f_2\}, \{c_1, c_2\} \rangle,$
  $\langle \{f_2\}, \{c_1, c_2\} \rangle, \langle \{f_1\}, \{c_1, c_2\} \rangle$
- removing unreasonable yields

$$\langle \{f_1\}, \{c_1\} \rangle, \langle \{f_1, f_2\}, \{c_1, c_2\} \rangle$$

Conclusions and Future Work

- Two-tiered formalism provides insight into the problematics.
- Resolves ambiguity. When explaining your approach, think of the problem-solution pairs allowed.

- How do languages used in practice map to our formalism?
- Would it be useful to have a language construct "realized-by"?

Bonus Slide

Interpretation 3 different than 2

$$\begin{aligned} \neg(f_1 \wedge f_2) \wedge \\ c_2 \Rightarrow c_1 \wedge \\ f_1 \text{ realized-by } c_1 \wedge \\ f_2 \text{ realized-by } c_2 \end{aligned}$$

Works for non-boolean models

- introduce orderings $\sqsubseteq_{\mathrm{f}}$, $\sqsubseteq_{\mathrm{c}}$
- reasonable combinations as those that can't be improved
- $\langle \mathbf{f}, \mathbf{c} \rangle$ not improvable iff

$$(\forall \langle \mathbf{f}', \mathbf{c}' \rangle \in \mathcal{M}_{\mathrm{fc}})((\mathbf{f} \sqsubseteq_{\mathrm{f}} \mathbf{f}' \wedge \mathbf{c}' \sqsubseteq_{\mathrm{c}} \mathbf{c}) \Rightarrow (\mathbf{f} = \mathbf{f}' \wedge \mathbf{c} = \mathbf{c}'))$$