Formal Approach to Integrating Feature and Architecture Models

Mikoláš Janota[1] Goetz Botterweck[2]

Lero
[1]University College Dublin
[2]University of Limerick
Ireland

ITU August 2008

IST-15905

Introduction

Software Product Lines

- development of families of similar systems
- systematically exploit commonality in the family
- *explicit* modeling of the family

Introduction

Software Product Lines

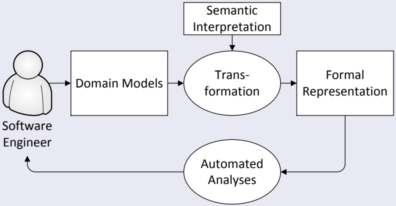
- development of families of similar systems
- systematically exploit commonality in the family
- *explicit* modeling of the family

Modeling

- feature models are customer oriented
- architecture models are solution oriented
- our work provides formal foundation for integrating the two

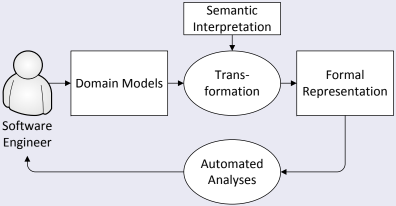
Why Formalize?

Formalisms applied in domain engineering



Why Formalize?

Formalisms applied in domain engineering

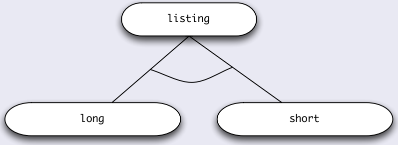


Formalisms in research

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

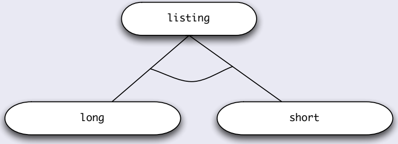
Feature Models and their Semantics

Example



Feature Models and their Semantics

Example

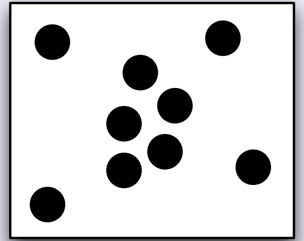


- semantics as allowed combinations

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

Generalization

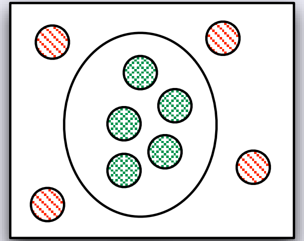
Domain



Problem space

Generalization

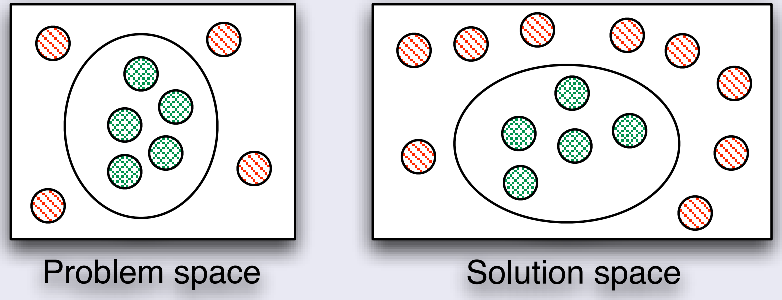
Domain model



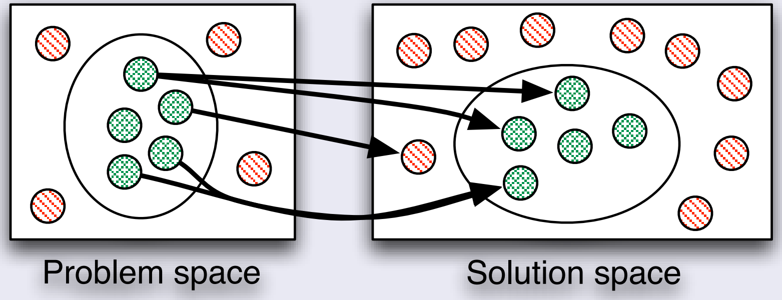
Problem space

Generalization

Domain and Solution model



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 combinations:

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

Applications

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

Examples

- implementable feature combinations:

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

- Is the original feature model OK?

$$\mathcal{F} \subseteq \mathcal{I}_{\mathcal{M}_{\mathrm{fc}}}$$

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

$$
\begin{aligned}
f_1 \vee f_2 &\wedge \\
c_2 \Rightarrow c_1 &\wedge \\
f_1 \text{ realized-by } c_1 &\wedge \\
f_2 \text{ realized-by } c_2 &
\end{aligned}
$$

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

- remove combinations that have unnecessary components or features that are not selected but are implemented
- implication interpretation (1)

$$\begin{array}{ll} \langle\{f_1\}, \{c_1\}\rangle & \langle\{f_1\}, \{c_1, c_2\}\rangle \\ \langle\{f_1, f_2\}, \{c_1, c_2\}\rangle & \langle\{f_2\}, \{c_1, c_2\}\rangle \end{array}$$

- removing unreasonable combinatins yields (3)

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

Is the Interpretation 3 Different from 2?

Example

$$\begin{aligned} f_1 \text{ XOR } f_2 \wedge \quad & f_1 \text{ realized-by } c_1 \wedge \\ c_2 \Rightarrow c_1 \wedge \quad & f_2 \text{ realized-by } c_2 \end{aligned}$$

Is the Interpretation 3 Different from 2?

Example

$$\begin{array}{rl} f_1 \text{ XOR } f_2 \wedge & f_1 \text{ realized-by } c_1 \wedge \\ c_2 \Rightarrow c_1 \wedge & f_2 \text{ realized-by } c_2 \end{array}$$

Interpretation 1 (using $\Rightarrow$)

$$\begin{array}{rl} f_1 \text{ XOR } f_2 \wedge & f_1 \Rightarrow c_1 \wedge \\ c_2 \Rightarrow c_1 \wedge & f_2 \Rightarrow c_2 \end{array}$$

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

Is the Interpretation 3 Different from 2?

Example

$$\begin{aligned} f_1 \text{ XOR } f_2 \wedge &\quad f_1 \text{ realized-by } c_1 \wedge \\ c_2 \Rightarrow c_1 \wedge &\quad f_2 \text{ realized-by } c_2 \end{aligned}$$

Interpretation 1 (using $\Rightarrow$)

$$\begin{aligned} f_1 \text{ XOR } f_2 \wedge &\quad f_1 \Rightarrow c_1 \wedge \\ c_2 \Rightarrow c_1 \wedge &\quad f_2 \Rightarrow c_2 \end{aligned}$$

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

Interpretation 3 ($\Rightarrow$ − improvable)

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

Is the Interpretation 3 Different from 2?

Example

$$\begin{array}{rl} f_1 \text{ XOR } f_2 \wedge & f_1 \text{ realized-by } c_1 \wedge \\ c_2 \Rightarrow c_1 \wedge & f_2 \text{ realized-by } c_2 \end{array}$$

Interpretation 1 (using $\Rightarrow$)

$$\begin{array}{rl} f_1 \text{ XOR } f_2 \wedge & f_1 \Rightarrow c_1 \wedge \\ c_2 \Rightarrow c_1 \wedge & f_2 \Rightarrow c_2 \end{array}$$

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

Interpretation 3 ($\Rightarrow$ – improvable)

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

Interpretation 2 (using $\Leftrightarrow$)

$$\begin{array}{rl} f_1 \text{ XOR } f_2 \wedge & f_1 \Leftrightarrow c_1 \wedge \\ c_2 \Rightarrow c_1 \wedge & f_2 \Leftrightarrow c_2 \end{array} \qquad \{\langle\{f_1\}, \{c_1\}\rangle\}$$

Computing Interpretation 3 for Booleans

Conversion to a known problem

- Flip the meaning of component variables, i.e. a component C_i is in the configuration *iff* c_i.
- A configuration is not improvable *iff* it is *maximal* in the subset ordering.
- Known under the name *maximal models of a formula*.

Properties of maximal models

- checking maximality of a model is co-NP complete for CNF
- generating all maximal models cannot be done in an output-polynomial time even for Horn formulas

[Kavvadias *et al.*, Lonc and Truszczyński]

Interpretation 3 Formally

Orderings as a preference

- introduce *orderings* $\sqsubseteq_{\mathrm{f}}$, $\sqsubseteq_{\mathrm{c}}$
- reasonable combinations are those that *cannot be improved*
- $\langle \mathbf{f}, \mathbf{c} \rangle$ cannot be improved *iff*

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

Examples

- for boolean case declare:

  $$\sqsubseteq_{\mathrm{f}} \equiv \subseteq$$
  $$\sqsubseteq_{\mathrm{c}} \equiv \subseteq$$

- in more complicated cases:

  $$\mathbf{f}_1 \sqsubseteq_{\mathrm{f}} \mathbf{f}_2 \text{ \textit{iff} } \mathrm{performance}(\mathbf{f}_1) \leq \mathrm{performance}(\mathbf{f}_2)$$
  $$\mathbf{c}_1 \sqsubseteq_{\mathrm{c}} \mathbf{c}_2 \text{ \textit{iff} } \mathrm{price}(\mathbf{c}_1) \leq \mathrm{price}(\mathbf{c}_2)$$

Conclusions and Future Work

- Two-tiered formalism provides refined view on the problem.
- 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"?

- more information at

  `kind.ucd.ie`