Towards an Automated Fault Localizer while Designing Meta-models

Adel Ferdjoukh and Jean-Marie Mottu

MDEbug@MODELS 2018, Copenhagen (København)

TU WIEN

UNIVERSITÉ DE NANTES

Synopsis

1. Motivation
2. Automated fault localization
3. Tooling
4. Ideas for improving
5. Conclusion

1. Motivation

Validity of meta-models

GENERAL IDEA

Ensure the validity of meta-models & help meta-model designers

WISHED FEATURES

1. Localize problems in faulty meta-models
2. Give feedback to designers
3. Propose corrections

meta-model validation

Validity of meta-models

Generate valid instances using model finders

Characteristics of model finders

1. Automated
2. Many models
3. Meaningful and Diverse

Validation with Grimm

Grimm

A tool for model generation and meta-model validation

Validation with Grimm

1. Design a new meta-model
2. Ask for instances
3. Inspect and correct
4. Back to 2

Validation with Grimm

Grimm

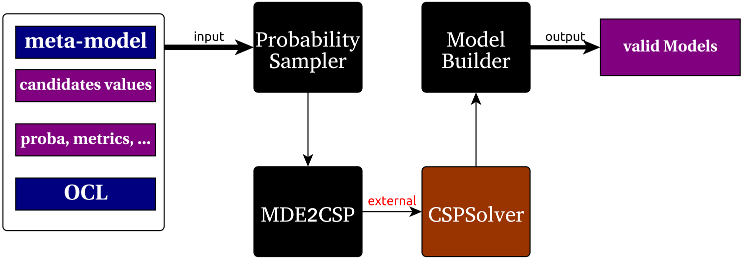
A tool for model generation and meta-model validation

Validation with Grimm

1. Design a new meta-model
2. Ask for instances
3. Inspect and correct (manual task)
4. Back to 2

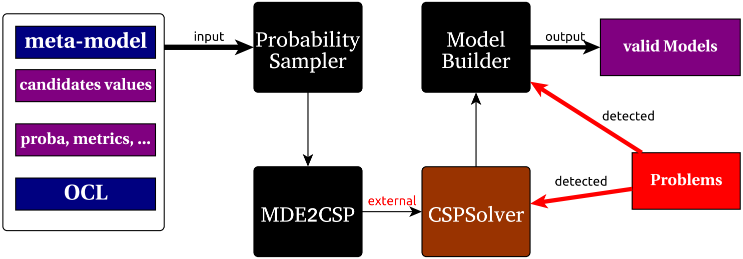
Steps for model generation

STEPS FOR MODEL GENERATION



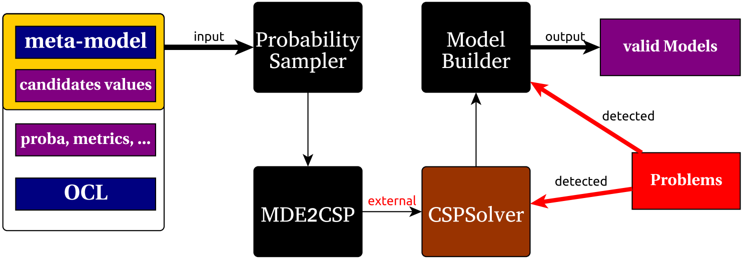
Steps for model generation

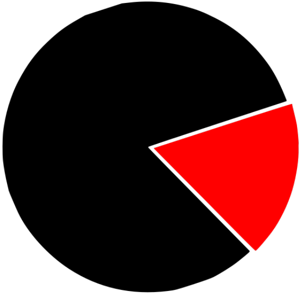
DETECTION FAR FROM ORIGIN



Steps for model generation

FOCUS FOR CURRENT WORK





2. Automated fault localization

Proposition

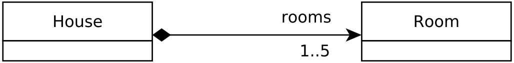
Automated fault localization

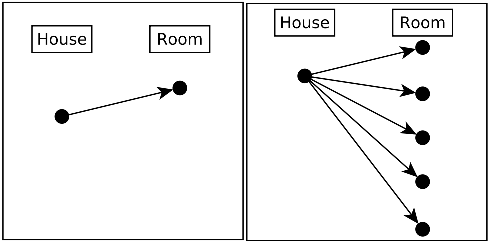
- Static analysis of meta-model
- Check the consistency of generation parameters
- Precise localization of errors

Systems of Linear Inequalities

- Translate a meta-model and generation parameters into SLI
- Write checking algorithms
- Give fixing propositions

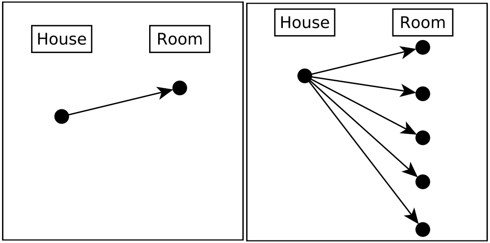
From Ecore to SLI





$$\begin{cases} \#House \leq \#Room \\ \#Room \leq 5 \times \#House \end{cases}$$

From Ecore to SLI



$$\begin{cases} \#House \leq \#Room \\ \#Room \leq 5 \times \#House \end{cases}$$

- 3H & 4R
- 2H & 1R

From Ecore to SLI

Translated Ecore elements

1. Simple references
2. Compositions
3. Eopposite references
4. Inheritance combined with 1,2 and 3

Checking the SLI and propositions

Checking the SLI

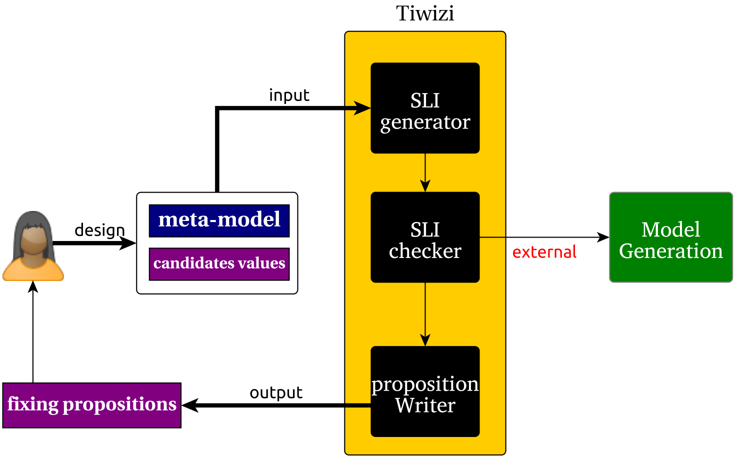
- Each Inequality is checked using the candidate values
- Detect all faults in 1 shot

Fixing propositions

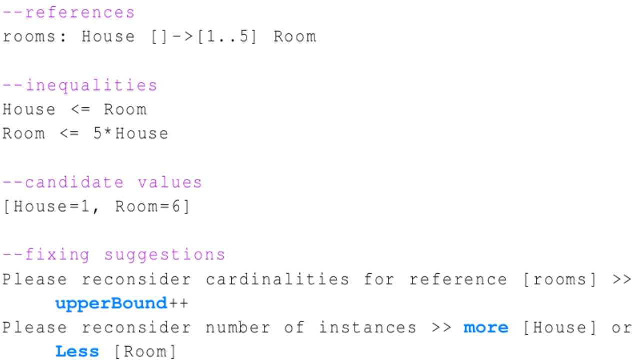
- Detected anomaly
- Help propositions (manual fixing)

3. Tooling

Automated Fault Localization



TIWIZI tool



```
--references
rooms: House []->[1..5] Room

--inequalities
House <= Room
Room <= 5*House

--candidate values
[House=1, Room=6]

--fixing suggestions
Please reconsider cardinalities for reference [rooms] >>
    upperBound++
Please reconsider number of instances >> more [House] or
    Less [Room]
```

4. Ideas for improving

Meta-model & partial solutions

Meta-model validity

- Solve the SLI to propose solutions (intervals of values)
- Detect meta-model anomalies

Partial solutions

1. Users give some CVs (not all classes)
2. Solve the SLI
3. Complete remaining CVs

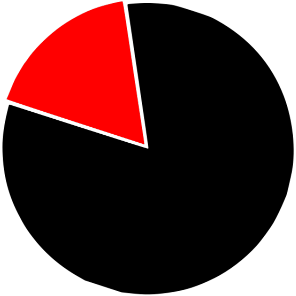
Global fixing propositions

Current solution

Fixing propositions concern one reference (or two classes).

Improvement idea

- Learn more complex propositions (eg. 3 classes at once)
- Auto-fix generation



5. Conclusion

Conclusion

Summary

Approach for assisting meta-model designers

Contributions

- Translation of Ecore meta-models into SLI
- Automated fault localization during instantiation.
- Precise fixing propositions

Tanmirt

ⵜⴰⵏⵎⵉⵔⵜ