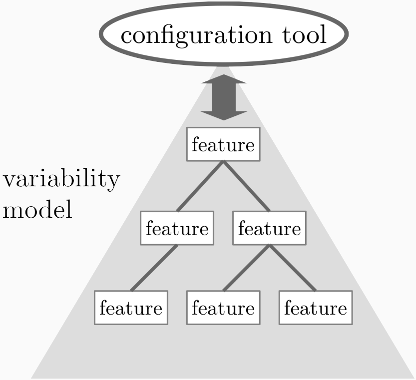
On the Relation of Variability Modeling Languages and Non-Functional Properties

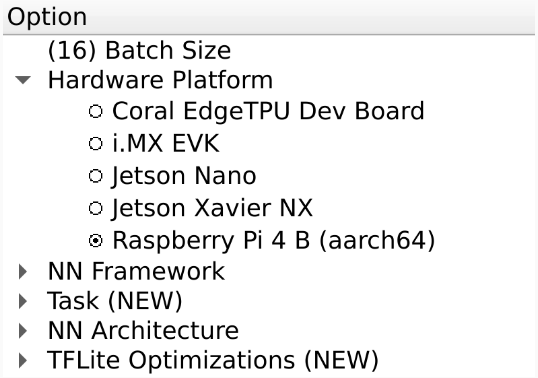
Birte Friesel, Michael Müller, Matheus Ferraz, Olaf Spinczyk birte.friesel@uos.de

September 13, 2022

Universität Osnabrück / Arbeitsgruppe Eingebettete Softwaresysteme

Variability Modeling Languages





Variability models → interactive software product line configuration

Non-Functional Properties

Configuration

Cost: 60 € | Inference Time: 532 ms

Memory Footprint: 122 MB | Model Size: 4 MB | **Throughput: 32.1 FPS**

Batch Size 16

Hardware Platform

NN Framework

Task (NEW)

NN Architecture

Quantization (NEW)

hw_platform

Coral EdgeTPU Dev Board ☐ +100 € -12.4 FPS

i.MX EVK ☐ +440 € +88.5 FPS

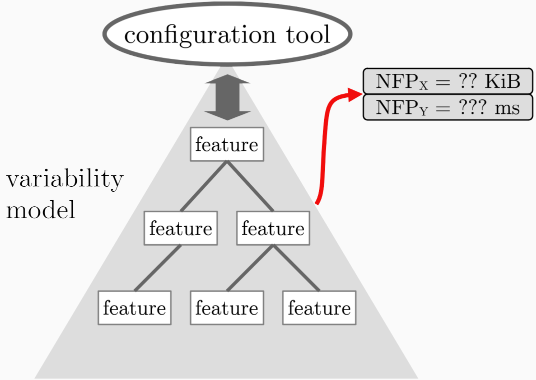
Jetson Nano ☐ +57 € -1.2 FPS

Jetson Xavier NX ☐ +420 € +14.2 FPS

Raspberry Pi 4 B (aarch64) ☑

Non-functional property (NFP) models → performance-aware configuration

On the Relation of ...



How to add non-functional properties to a variability model?

Contents

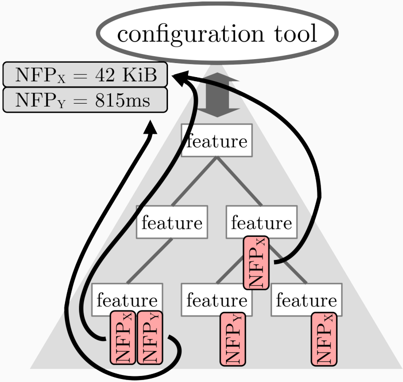
1 Approaches

2 Analysis

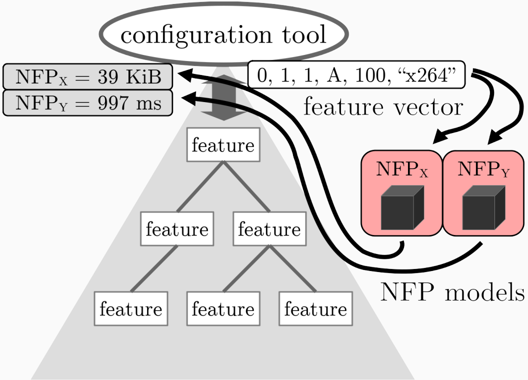
3 Evaluation

4 Conclusion

Approaches



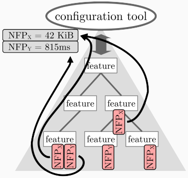
Integrated NFP Model



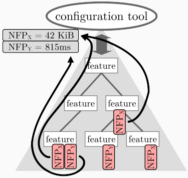
Separate NFP Model

Integrated NFP Model

- NFPs are **attributes** of individual **features**
- Aggregation functions define NFP of the complete product



Integrated NFP Model



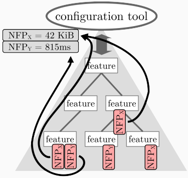
- NFPs are **attributes** of individual **features**
- Aggregation functions define NFP of the complete product

E.g. ClaferMoo [Ola+12]

```
EdgeTPU : HWFeature [cost = 160]
RasPi4  :  HWFeature [cost = 60]
Battery :  HWFeature [cost = 50]

totalCost :  integer [ totalCost = sum HWFeature.cost ]
```

Integrated NFP Model



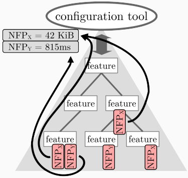
- NFPs are **attributes** of individual **features**
- Aggregation functions define NFP of the complete product

E.g. TVL [Bou+10]

```
EdgeTPU {int cost is 160;}
RasPi4  {int cost is 60;}
Battery {int cost is 50;}

EdgeML {int cost is sum(selectedChildren.cost)}
```

Integrated NFP Model



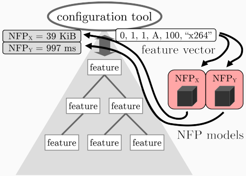
- NFPs are **attributes** of individual **features**
- Aggregation functions define NFP of the complete product

E.g. UVL [Sun+21]

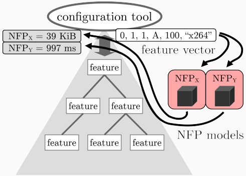
```
EdgeTPU {cost 160}
RasPi4  {cost 60}
Battery {cost 50}
```

Separate NFP Model

- **Feature vector** $\vec{x}$ describes product configuration
- Calculate NFP y using separate model function $f : \vec{x} \mapsto y$



Separate NFP Model



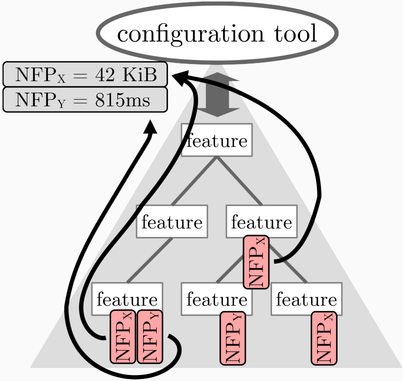
- **Feature vector** $\vec{x}$ describes product configuration
- Calculate NFP y using separate model function $f : \vec{x} \mapsto y$

E.g. $\vec{x} = (x_{\text{HW}}, x_{\text{Bat}}) \in \{\{\text{EdgeTPU}, \text{RasPi4}\}, \{0, 1\}\}$

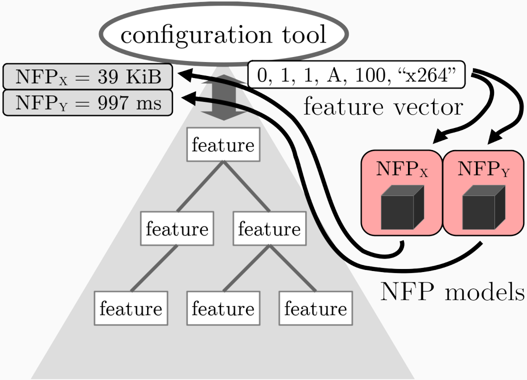
$$cost(\vec{x}) = 50 \cdot x_{\text{Bat}} + \begin{cases} 160 & x_{\text{HW}} = \text{EdgeTPU} \\ 60 & x_{\text{HW}} = \text{RasPi4} \end{cases}$$

Arbitrary functions can be used, e.g. regression trees or neural networks

Approaches



Integrated NFP Model



Separate NFP Model

Should NFP models be part of the variability model?

Contents

1 Approaches

2 Analysis

3 Evaluation

4 Conclusion

Analysis

Annotation / Training → NFP Model → Insights / Prediction

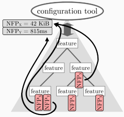
Insights / Prediction: Domain Expert, Algorithm

Annotation Process

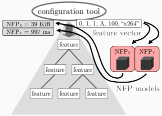
Complexity
Expressiveness
Maintenance
Modularity

Accuracy

Annotation Process



Integrated NFP Model

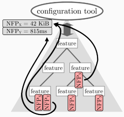


Separate NFP Model

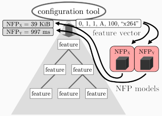
- Manual annotation
- Benchmarks → model training

- Manual annotation
- Benchmarks → model training

Annotation Process



Integrated NFP Model



Separate NFP Model

✓ Manual annotation

(✓) Benchmarks → model training

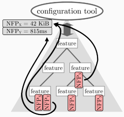
```
size = sum feat.size
∗ (debug ?  1.2 :  1)
```

(✓) Manual annotation

✓ Benchmarks → model training

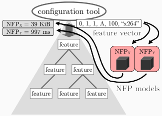
E.g. CART, XGBoost, neural networks

Annotation Process



Integrated NFP Model

≈



Separate NFP Model

✓ Manual annotation

(✓) Benchmarks → model training

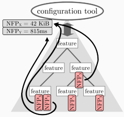
```
size = sum feat.size
∗ (debug ?  1.2 :  1)
```

(✓) Manual annotation

✓ Benchmarks → model training

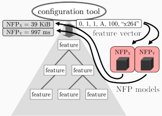
E.g. CART, XGBoost, neural networks

Expressiveness



Integrated NFP Model

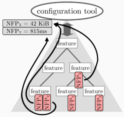
- Defined by modeling language
- Typically limited to
  - feature-wise annotations
  - feature interaction
  - aggregate functions



Separate NFP Model

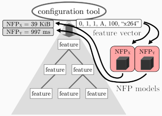
- Chosen as suitable
- Near arbitrary, e.g.
  - feature-wise annotations
  - regression trees
  - neural networks

Expressiveness



Integrated NFP Model

<



Separate NFP Model

- Defined by modeling language
- Typically limited to
  - feature-wise annotations
  - feature interaction
  - aggregate functions

- Chosen as suitable
- Near arbitrary, e.g.
  - feature-wise annotations
  - regression trees
  - neural networks

Cross-Cutting Concerns

Features may interact with each other (→ not independent)

DB		**Debug**	**Safety**
–	172 kB	+ 45 kB	+ 18 kB
Multi	+ 20 kB		
WAL	+ 32 kB		

Cross-Cutting Concerns

Features may interact with each other (→ not independent)

DB		Debug	Safety
–	172 kB	+ 45 kB	+ 18 kB
Multi	+ 20 kB	+ 3 kB	+ 11 kB
WAL	+ 32 kB	+ 4 kB	+ 0 kB

Cross-Cutting Concerns

Features may interact with each other (→ not independent)

DB		Debug	Safety
–	172 kB	+ 45 kB	+ 18 kB
Multi	+ 20 kB	+ 3 kB	+ 11 kB
WAL	+ 32 kB	+ 4 kB	+ 0 kB

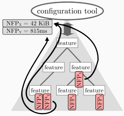
- Handled in variability modeling languages by *feature interaction* [Sie+12]
  - Check each feature pair A, B for interaction (domain expert or benchmarks)
  - If yes: add feature AB with $AB \Leftrightarrow A \wedge B$ to variability model
  - E.g.: (Multi, Debug) = 3 kB; (Multi, Safety) = 11 kB; (WAL, Debug) = 4 kB
  - Can be extended for more complex interactions (e.g. $ABCD$)

Features may interact with each other (→ not independent)

DB		Debug	Safety
–	172 kB	+ 45 kB	+ 18 kB
Multi	+ 20 kB	+ 3 kB	+ 11 kB
WAL	+ 32 kB	+ 4 kB	+ 0 kB

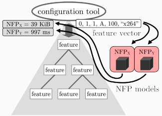
- Handled in variability modeling languages by *feature interaction* [Sie+12]
  - Check each feature pair A, B for interaction (domain expert or benchmarks)
  - If yes: add feature AB with $AB \Leftrightarrow A \land B$ to variability model
  - E.g.: (Multi, Debug) = 3 kB; (Multi, Safety) = 11 kB; (WAL, Debug) = 4 kB
  - Can be extended for more complex interactions (e.g. $ABCD$)
- Separate NFP models can automatically learn about this

Complexity



Integrated NFP Model

>

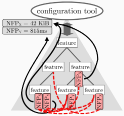


Separate NFP Model

- Feature-wise annotations: simple → easy to understand

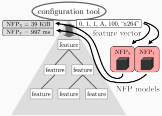
- Depends on model type
- XGBoost, NN: hard to grasp

Complexity



Integrated NFP Model

≈

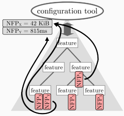


Separate NFP Model

- Feature-wise annotations: simple → easy to understand
- Feature interactions clutter the model

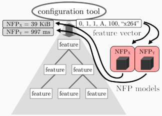
- Depends on model type
- XGBoost, NN: hard to grasp
- Regression model trees: Expressive and understandable [FS22]

Maintenance and Modularity



Integrated NFP Model

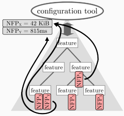
- Method defined by variability modeling language
- No separation of concerns: NFP attributes become useless after implementation changes



Separate NFP Model

- Method can be changed at will
- Implementation change → new NFP model or transfer learning [Jam+18]

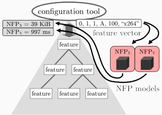
Maintenance and Modularity



Integrated NFP Model

- Method defined by variability modeling language
- No separation of concerns: NFP attributes become useless after implementation changes

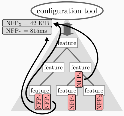
<



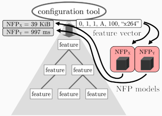
Separate NFP Model

- Method can be changed at will
- Implementation change → new NFP model or transfer learning [Jam+18]

Comparison



Integrated NFP Model



Separate NFP Model

- Annotation by domain expert
- Clear feature ↔ NFP relation
- cross-cutting concerns present
  → inaccurate or complex

- Automated generation
- Separation of concerns
- Arbitrary model complexity
  → problem-specific approaches

Contents

1 Approaches

2 Analysis

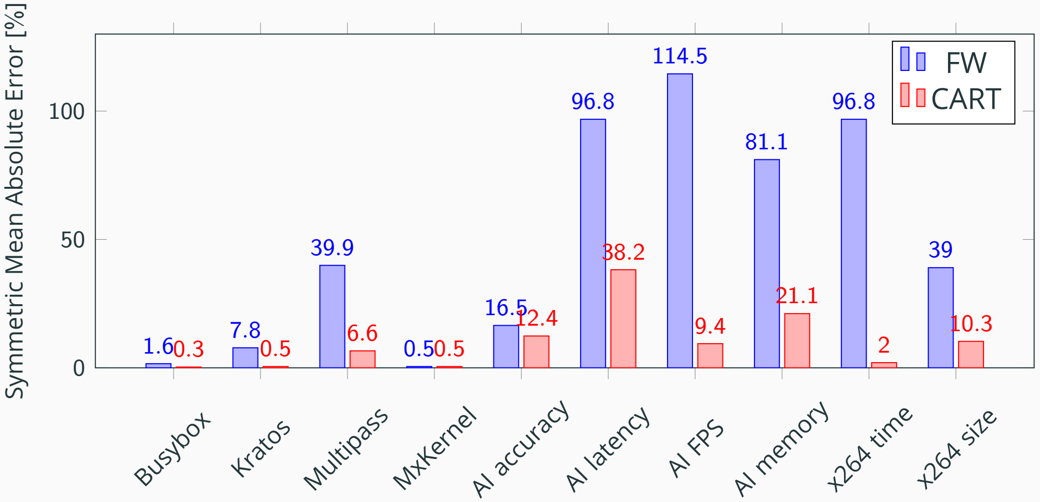
3 Evaluation

4 Conclusion

Evaluation Setup

- Integrated model: Feature-wise annotations (FW)
  $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$
- Separate model: Classification and Regression Trees (CART) [Bre+84]
  Go-to approach for data-efficient NFP model generation [Guo+18]
- Six product lines:
  - **busybox** multi-call binary → Binary size
  - **Kratos**, **Multipass**, **MxKernel** research OSes → ROM usage
  - **resKIL** embedded AI product line → accuracy, latency, throughput, memory
  - **x264** video codec → encoding duration and file size

Model Error (10-fold cross validation)



Advantages of External Models

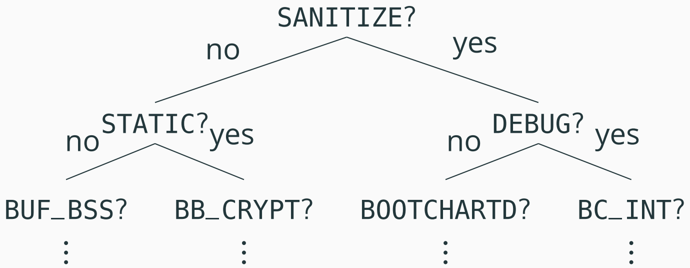
- Decision tree structure naturally captures dependencies between features

→ Higher model accuracy

Advantages of External Models

- Decision tree structure naturally captures dependencies between features

→ Higher model accuracy



- Influential features located close to the root

→ Understandable performance model

Contents

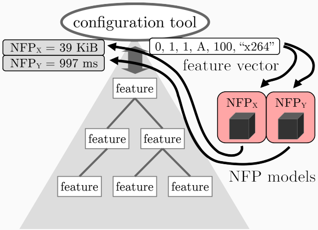
1 Approaches

2 Analysis

3 Evaluation

4 Conclusion

Conclusion



Separate NFP Model

- Opinion: variability models should **not** incorporate NFP-related concerns
- Instead:
  - Formalize configurations / products as **feature vectors**
  - Use configuration tool to link variability and NFP models

References i

[Bou+10] Quentin Boucher et al. **"Introducing TVL, a text-based feature modelling language".** In: Proceedings of the Fourth International Workshop on Variability Modelling of Software-intensive Systems (VaMoS'10). 2010, pp. 159–162.

[Bre+84] Leo Breiman et al. **Classification and regression trees.** Routledge, 1984. DOI: 10.1201/9781315139470.

[FS22] Birte Friesel and Olaf Spinczyk. **"Regression Model Trees: Compact Energy Models for Complex IoT Devices".** In: Proceedings of the Workshop on Benchmarking Cyber-Physical Systems and Internet of Things. CPS-IoTBench '22. IEEE, May 2022, pp. 1–6. DOI: 10.1109/CPS-IoTBench56135.2022.00007. URL: https://ess.cs.uos.de/static/videos/cpsiotbench22-Friesel-RMT.mp4.

References ii

[Guo+18] Jianmei Guo et al. **"Data-Efficient Performance Learning for Configurable Systems".** In: Empirical Softw. Engg. 23.3 (June 2018), pp. 1826–1867. ISSN: 1382-3256. DOI: `10.1007/s10664-017-9573-6`. URL: `https://doi.org/10.1007/s10664-017-9573-6`.

[Jam+18] Pooyan Jamshidi et al. **"Learning to Sample: Exploiting Similarities across Environments to Learn Performance Models for Configurable Systems".** In: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ESEC/FSE 2018. Lake Buena Vista, FL, USA: Association for Computing Machinery, 2018, pp. 71–82. ISBN: 9781450355735. DOI: `10.1145/3236024.3236074`. URL: `https://doi.org/10.1145/3236024.3236074`.

References iii

[Ola+12] Rafael Olaechea et al. **“Modelling and Multi-Objective Optimization of Quality Attributes in Variability-Rich Software”.** In: Proceedings of the Fourth International Workshop on Nonfunctional System Properties in Domain Specific Modeling Languages. NFPinDSML ’12. New York, NY, USA: Association for Computing Machinery, 2012. ISBN: 978-1-4503-1807-5. DOI: `10.1145/2420942.2420944`. URL: `https://doi.org/10.1145/2420942.2420944`.

[Sie+12] Norbert Siegmund et al. **“Predicting performance via automated feature-interaction detection”.** In: 2012 34th International Conference on Software Engineering (ICSE). 2012, pp. 167–177. DOI: `10.1109/ICSE.2012.6227196`.

References iv

[Sun+21] Chico Sundermann et al. **“Yet Another Textual Variability Language? A Community Effort towards a Unified Language”.** In: Proceedings of the 25th ACM International Systems and Software Product Line Conference - Volume A. New York, NY, USA: Association for Computing Machinery, 2021, pp. 136–147. ISBN: 9781450384698. URL: `https://doi.org/10.1145/3461001.3471145`.