Achieving Energy Efficiency with a Software Product Line Engineering Approach

Author Daniel Jesús Muñoz Guerra

Supervisors Lidia Fuentes Fernández

Mónica Pinto Alarcón

UNIVERSIDAD DE MÁLAGA

Why Energy Efficiency?

Why Green Software?

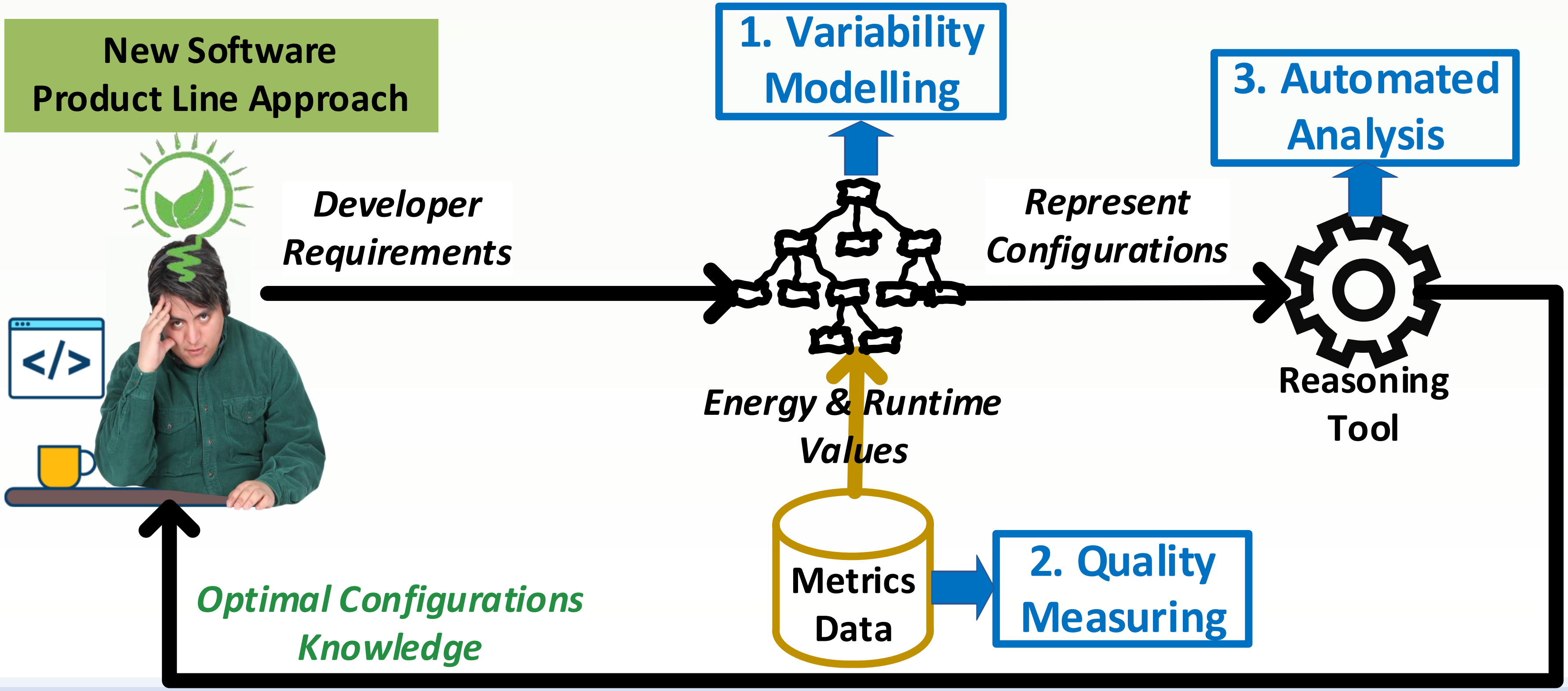
- E.g., Datacentres 1% of Global Energy
- ICT surpassed Aeronautic Industry
- Intel: "*We can reduce the software energy consumption up to a 90%*"
- Heterogenous technology: Cyber Physical Systems, the Cloud, Edge Computing, IoT, Virtual Networks…
- Wireless devices with large energy demands and low capacity batteries.

Objective:

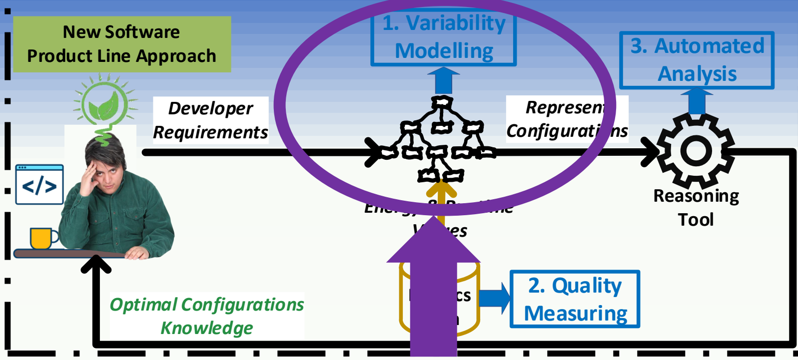
Minimise the Energy and Runtime

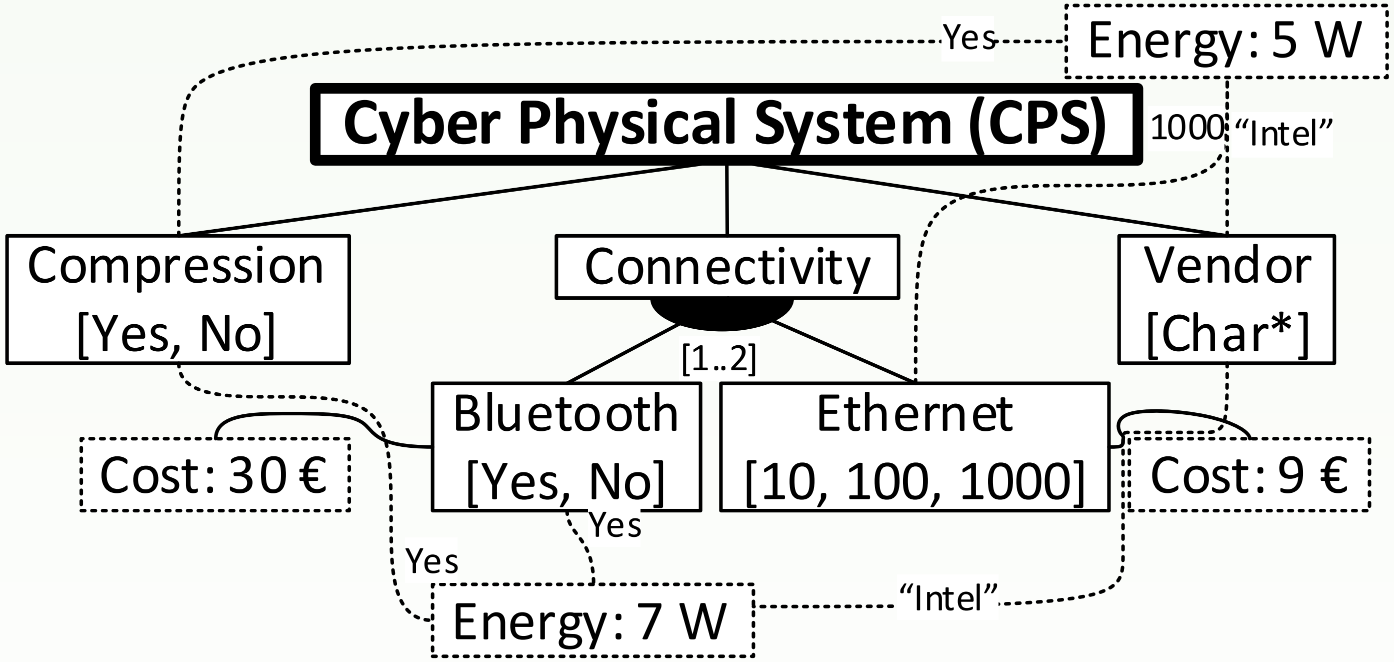
Automatically Analysing Alternative Features

of Complete Software Configurations



1. Variability Modelling

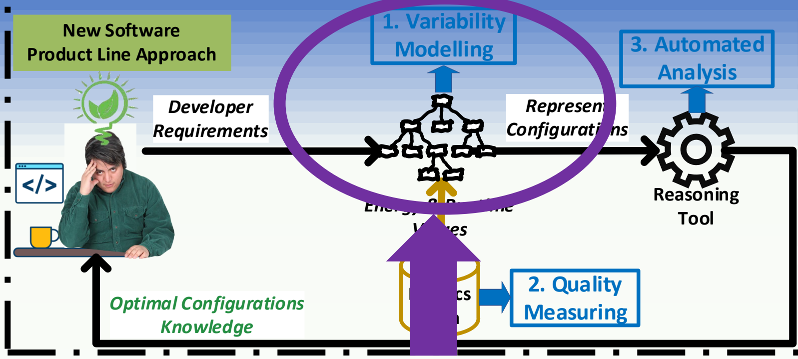


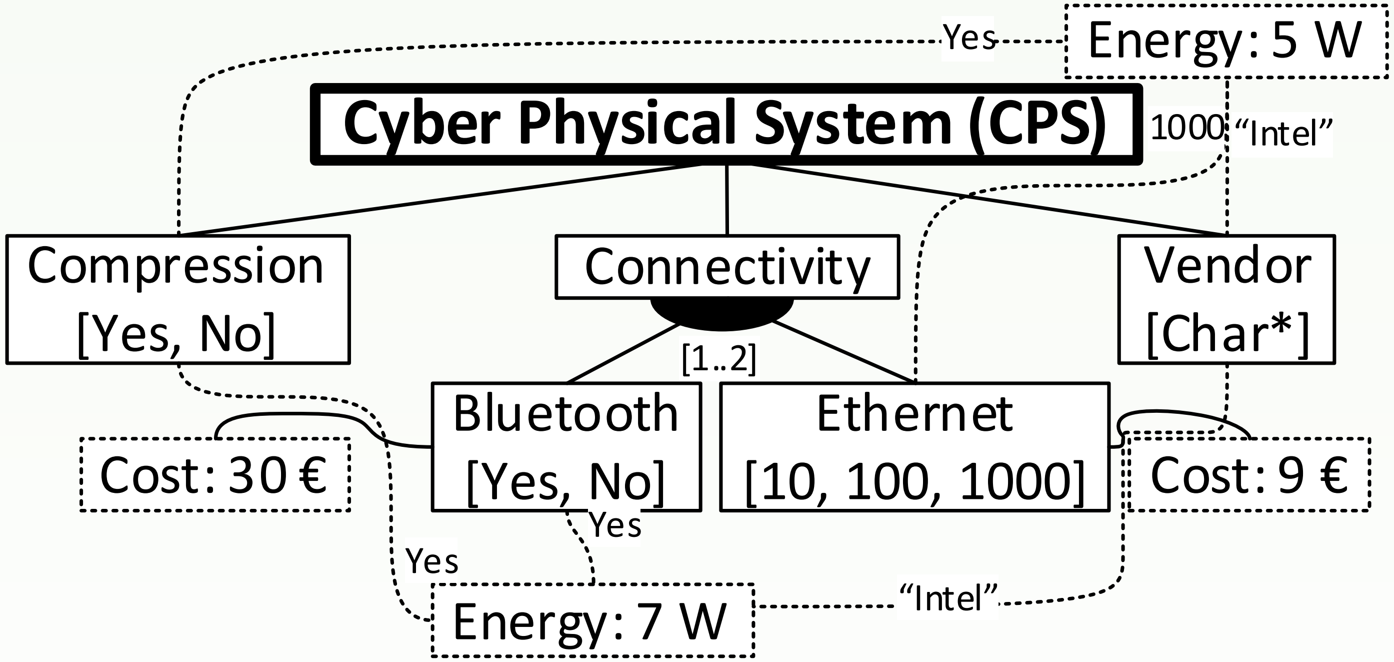


Very Simple Variability Model Example

- CPSs need different types of features: Boolean, Numerical and String
- CPSs have complex constraints:
  - E.g., Bluetooth and (Ethernet = 1000) requires (Vendor = "*(Media|Real)tek Inc.*")

1. Variability Modelling



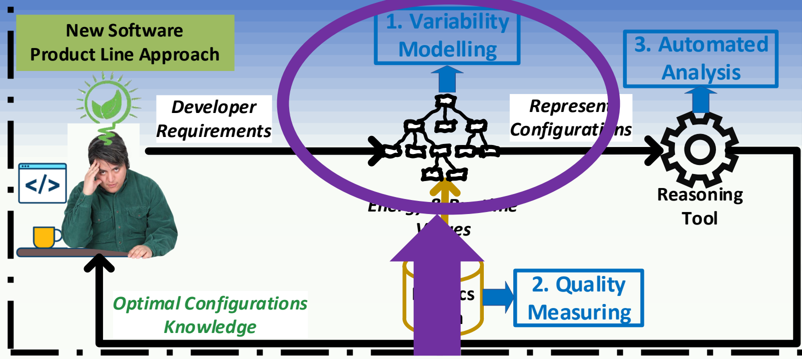


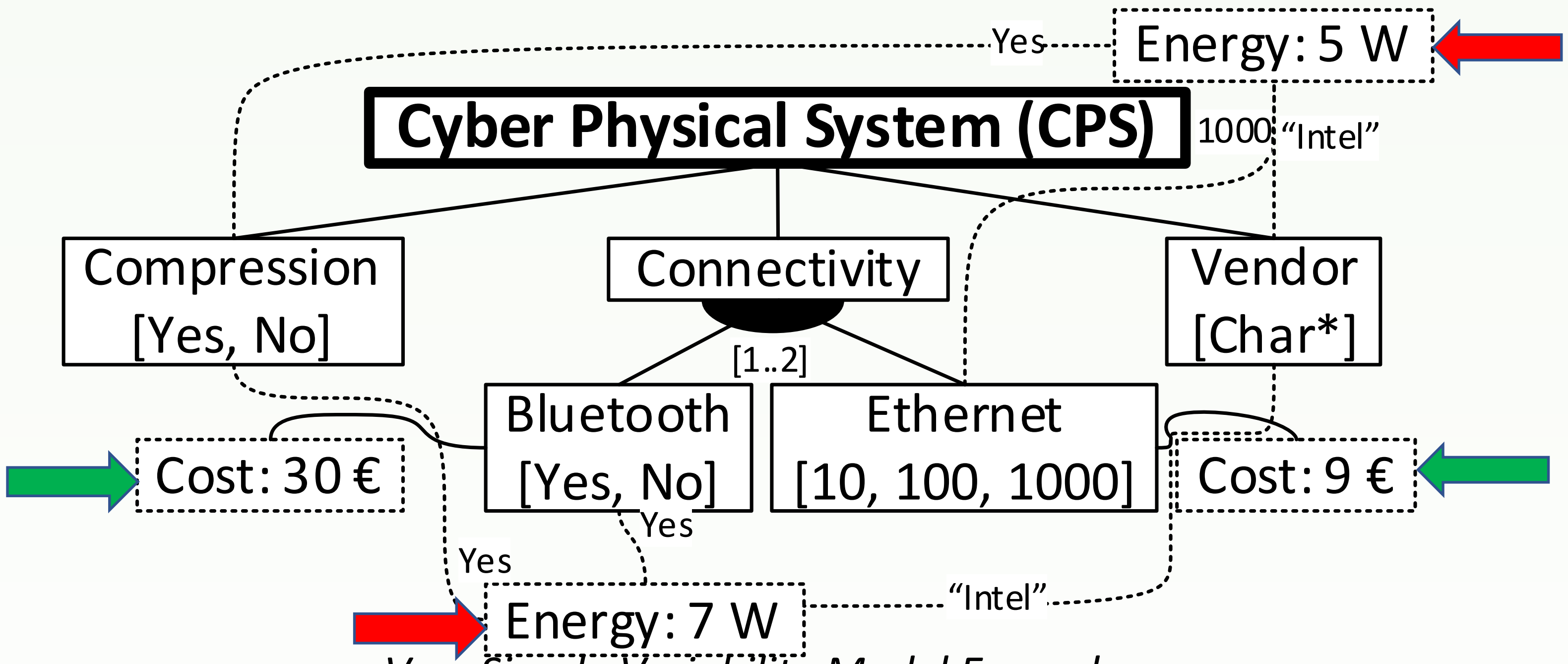
Very Simple Variability Model Example

- CPSs need different types of features: Boolean, Numerical and String
- CPSs have complex constraints:
  - E.g., Bluetooth and (Ethernet = 1000) requires (Vendor = "*(Media|Real)tek Inc.*")

Boolean Variability Models present many Limitations when Modelling CPSs

1. Variability Modelling



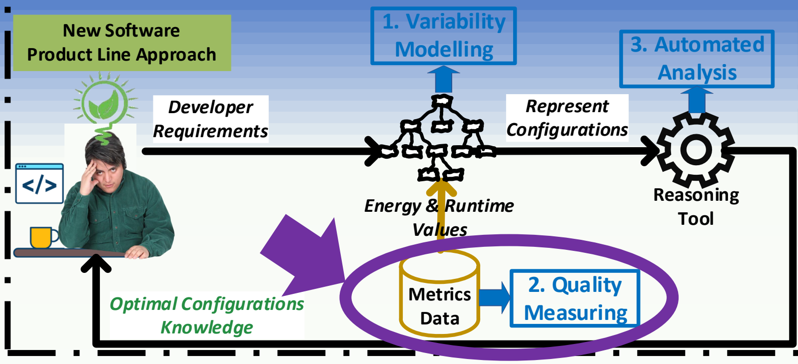


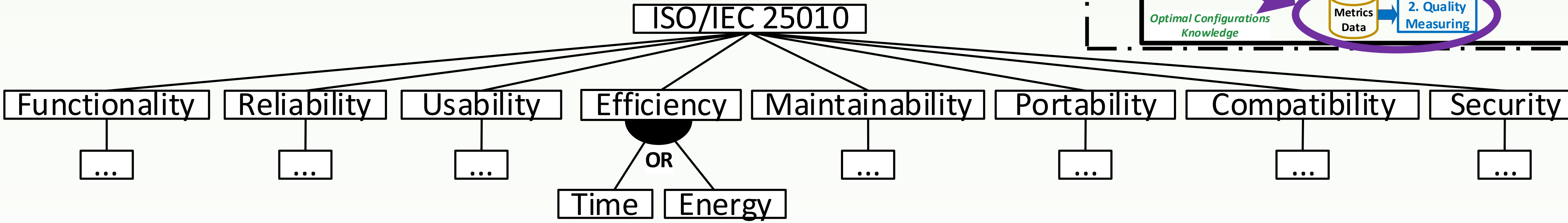
Very Simple Variability Model Example

- CPSs need different types of features: Boolean, Numerical and String
- CPSs have complex constraints:
  - E.g., Bluetooth and (Ethernet = 1000) requires (Vendor = "*(Media|Real)tek Inc.*")
- Qualities Attributes (QAs) interact at different levels: Feature/Variant-wise.

Feature-wise QAs CANNOT ACCURATELY MODEL configuration space QAs

2. Quality Measuring





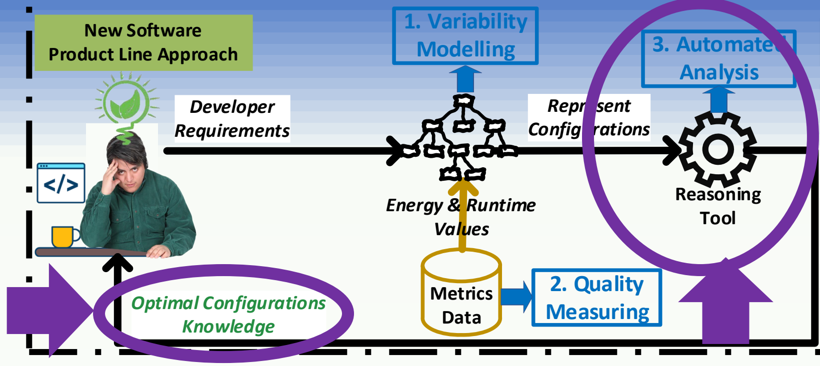
Quality Model: Standard ISO/IEC 25010

- **Energy consumption** is too complex for feature-wise modelling
  - **Quality**
  - Unfortu

Variant-wise Modelling and Measurement of Large Heterogeneous Configuration Spaces is hard

- Integrate variability, QAs modelling and measured values.
- We need tooling expertise to measure energy consumption.
- We cannot automatise the measurement of large heterogenous spaces.
  - E.g., Linux Kernel 2.6 configuration space size is $\sim 10^{2000}$ configurations.

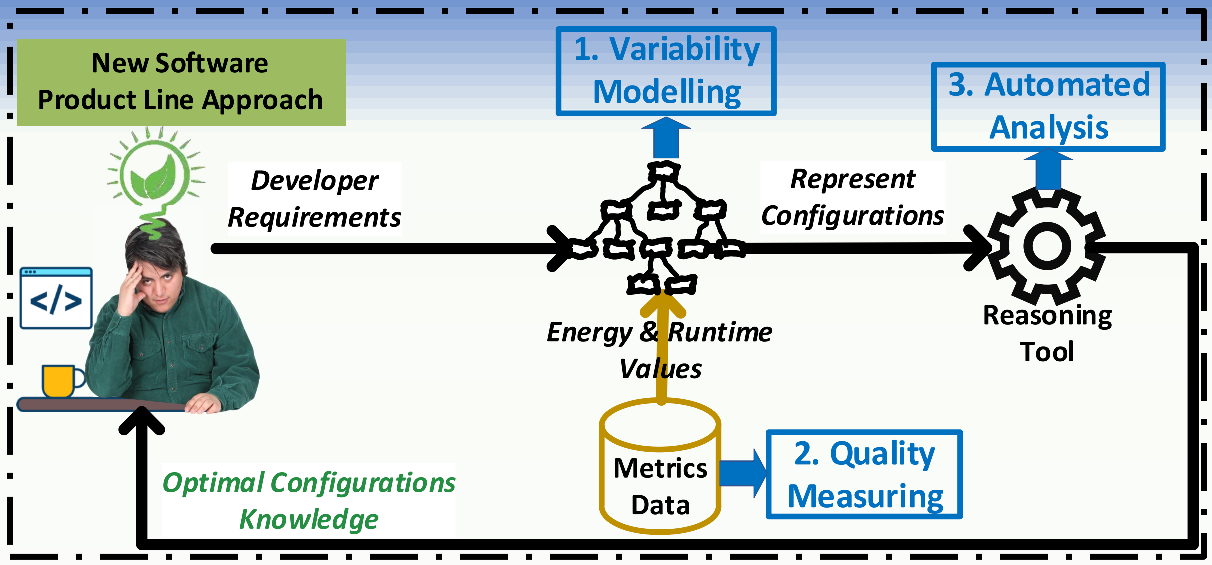
3. Automated Analyses



- An SPL approach provides the tools to analyse **large** variability spaces.
- We need many **operations**: satisfiability, counting, sampling, optimisation, etc.
- Different solvers for different problems **ordered by best** performing:
  - **#SAT** for counting Boolean SPLs.
  - **SAT**isfiability for Boolean SPLs.
  - **C**onstraint **P**rogramming for simple arithmetic SPLs.
  - **S**atisfiability **M**odulo **T**heory for complex arithmetic.

Not enough for real-world CPSs models as Variant-wise QAs are not natively supported

Research Questions

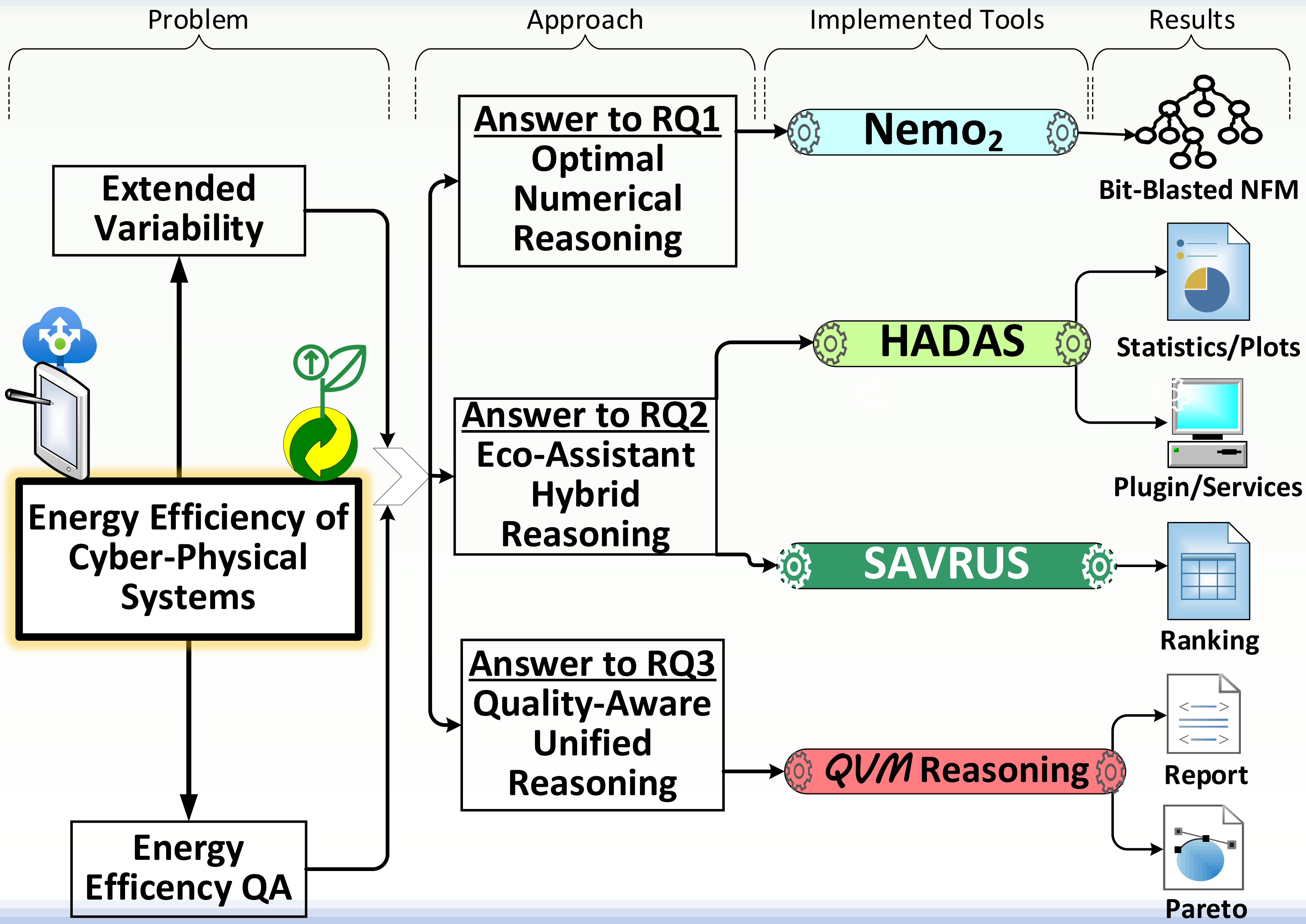


RQ1: How can we extend the state-of-the-art (Boolean) solvers to automatically support NFMs without performance degradation?

RQ2: How can we automatically provide energy efficiency insights when dealing with colossal, partially unknown, and partially-measured solution spaces based on NFMs and variant-wise QAs?

RQ3: How can we properly unify extended NFMs and QMs while having the native support of automated reasoning tools?

Big Picture


Problem
Approach
Implemented Tools
Results
Extended Variability
Energy Efficiency of Cyber-Physical Systems
Energy Efficency QA
Answer to RQ1
Optimal Numerical Reasoning
Answer to RQ2
Eco-Assistant Hybrid Reasoning
Answer to RQ3
Quality-Aware Unified Reasoning
Nemo$_2$
HADAS
SAVRUS
QVM Reasoning
Bit-Blasted NFM
Statistics/Plots
Plugin/Services
Ranking
Report
Pareto

BACKGROUND

Feature-Oriented Software Product Lines and Quality Attributes

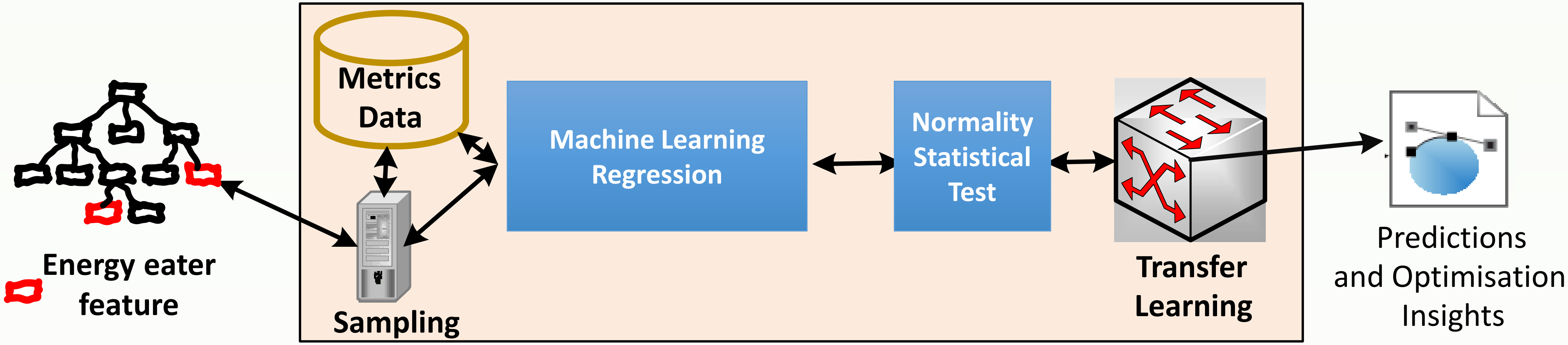
We need feature models that support numerical features, feature-wise QAs and variant-wise QAs - RQs[1-3]

- Classical **F**eature **M**odels (FMs): 1st formalisation of Boolean VMs.
- Numerical FMs: FMs that support arithmetic (e.g., Integer).
- Extended FMs: FMs that support feature-wise QAs with aggregation formulas.
  - E.g., Cost QA aggregation is a simple Addition of individual costs.
- Other representations of Boolean VMs
  - Propositional Formulas (PF)
  - Binary Decision Diagrams (BDD). They are the fastest for counting-based operations.

Quality Models are disconnected from Feature Models

Sampling, Statistics and Machine Learning for QAs

We need techniques that help to identify the energy eater features without measuring and analysing the complete configuration space



- Representative subspace by sampling: T-wise, Random, Recursive and Probabilistic - **RQs[1-3]**.
- Statistical significancy: x^2 and **M**ann-**W**hitney **U** (MWU) - **RQs[1-2]**.
- Regression Approximation of QA values: ***k*-N**earest **N**eighbourgs (*k*NN) - **RQ2[*SAVRUS*]**.
- Transfer Learning: Rank interactions by Scores function - **RQ2[*SAVRUS*]**.

RQ1 Approach
Optimal Numerical Reasoning

RQ1 Contribution 1

Bit-Blasting: Encode Numerical Features as **bits**, and arithmetic as PF clauses:

- Bit-vectors are signed integers by Big-endian binary two's complement encoding.
- Arithmetic: equality (**=**), inequalities (**≠,>, >**), addition (**+**), subtraction (**-**), multiplication (*****), division (**/**), and modulo (**%**).

Propositional Formulas for 3-bit Two's Complement Signed Integers

Operation	Bit-Blasted model	Propositional formula
$(NF_a \neq NF_b)$	$(a_1 \neq b_1) \vee (a_2 \neq b_2) \vee (a_3 \neq b_3)$	$(a_1 \oplus b_1) \vee (a_2 \oplus b_2) \vee (a_3 \oplus b_3)$
$(NF_a > NF_b)$	$(a_1 < b_1) \vee ((a_1 == b_1) \wedge (a_2 > b_2)) \vee ((a_1 == b_1) \wedge (a_2 == b_2) \wedge (a_3 > b_3))$	$(\neg a_1 \wedge b_1) \vee ((a_1 \Leftrightarrow b_1) \wedge (a_2 \wedge \neg b_2)) \vee ((a_1 \Leftrightarrow b_1) \wedge (a_2 \Leftrightarrow b_2) \wedge (a_3 \wedge \neg b_3))$

Munoz et al. – The Journal of Systems and Software 204 (2023) 111770

RQ1 Contribution 2

Definition of our own NFM language and support of the **fastest reasoning tools** formats: FMs in Universal Variability Language (**UVL**), **DIMACS** and **PF**.

```
def A bool 0     # 0 means new feature
def B bool B     # named in adjunt FM as B
def C bool 0
def D_unsigned [0:1]
def E_unsigned [0:3]
def F_signed [-1:1]
def G_enum_signed [-9, -3, 0, 3]
def H_constant [-2]
ct  C -> B
ct  A -> (G = 0)
ct  A or B
ct (G_enum_signed*H_constant) ≤ E_unsigned
```

Example of extending with Nemo2 an FM with new boolean and numerical features and constraints

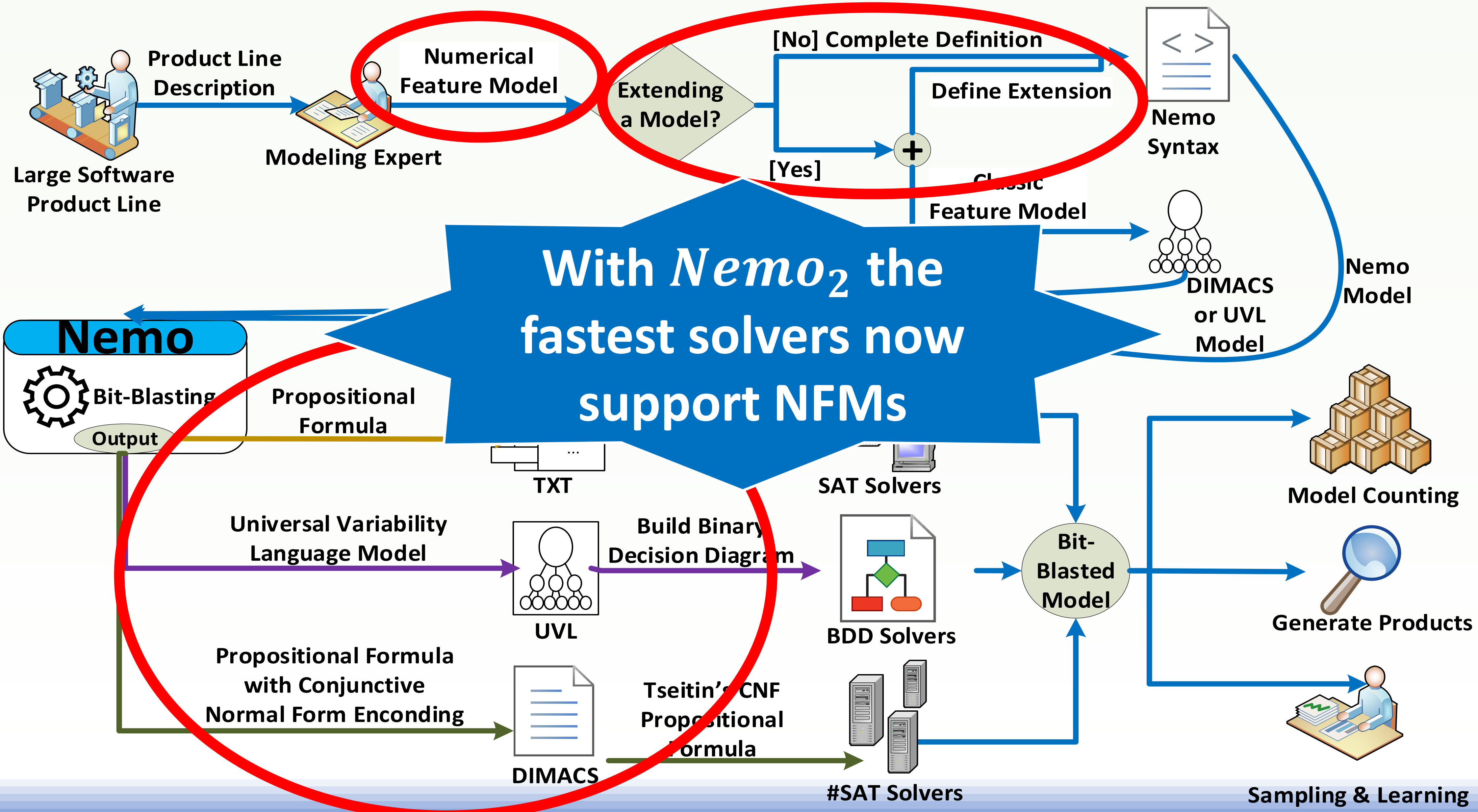
$NEMO_2$

```
features
  Root
    optional
      v1 # A_1
      v2 # A_2
      v3 # B_1
      v4 # B_2
      C  # Boolean
constraints
  !(!((v1 | !v2)))
  !(!((v2 | v1)))
  !(!((v3 | !v4)))
  (!(C) | !(!((v1 <=> v3 | !v2 <=> v4))))
```

Nemo2 UVL output for: A ∈ [-1,0]; B ∈ [-1,1] C Boolean in "C requires (A != B)"

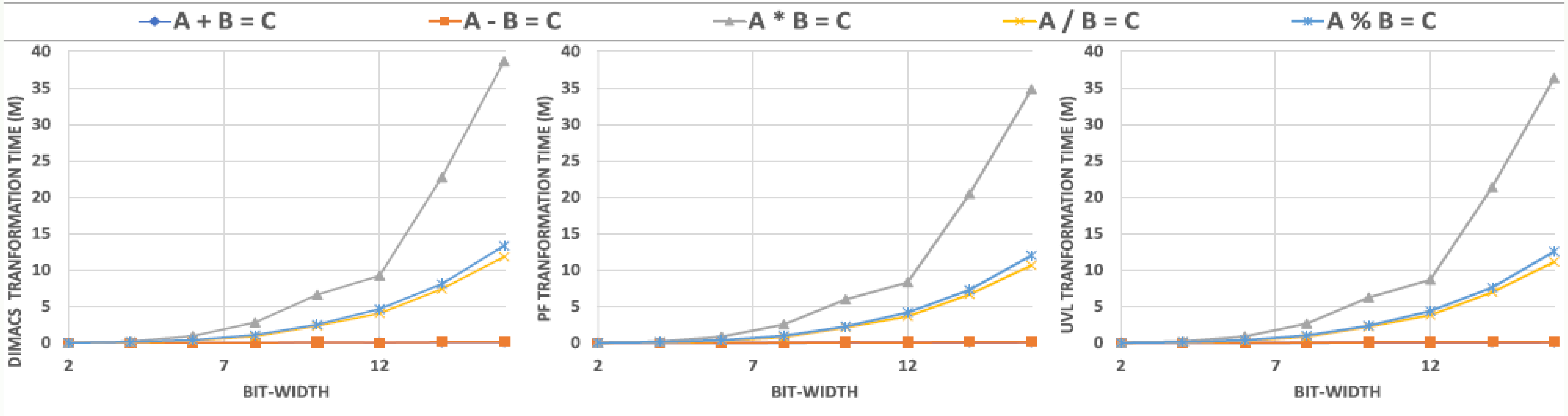
Munoz et al. – The Journal of Systems and Software 204 (2023) 111770

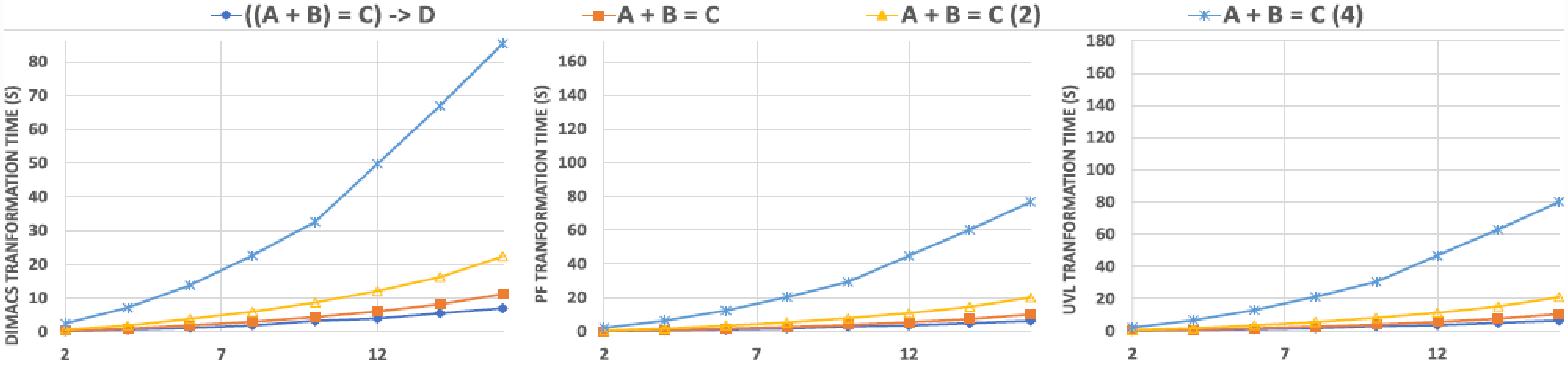
RQ1 Tool Support: $Nemo_2$ - https://github.com/danieljmg/Nemo2_tool



Munoz et al. – The Journal of Systems and Software 204 (2023) 111770

RQ1. Benchmarking $Nemo_2$ Scalability





Munoz et al. – The Journal of Systems and Software 204 (2023) 111770

$Nemo_2$ can <u>bit-blast</u> up to 12 bit-widths arithmetic operations in seconds

RQ1. Models Evaluated

NFM	Description	#F	#NFs	#Configs
Dune	Multi-grid solver	11	3	2,304
HSMGP	Stencil-grid solver	14	3	3,456
HiPAcc	Image processing	33	2	13,485
Trimesh	Triangle mesh library	13	4	239,360
MOTIV	Mobile Video Sequence	8	13	3.52×10^{30}
WeaFQAs	Quality Attributes Weaver	240	5	1.38×10^{40}
Fiasco	Real-time microkernel	234	5	3.06×10^{12}
axTLS	Client–server library	94	9	4.96×10^{38}
uClibc-ng	C Language library	269	6	8.20×10^{45}
Busybox 1.18.5	Embedded Linux	631	12	1.34×10^{191}
Busybox 1.28	Embedded Linux	1100	12	1.53×10^{248}
Linux 2.6.33.3	Operating System Kernel	6467	55	$\sim 5.66 \times 10^{1953}$

Munoz et al. – The Journal of Systems and Software 204 (2023) 111770

RQ1. Results and Answers

Model counting time					Sampling time	
Counting time	Glucose3	sharpSAT	Flamapy BDD	BDDSampler	Flamapy BDD	BDDSampler
Dune	0.37 s	0.01 s	0.03 s	0.44 s	2.79 s	3 s
HSMGP	69.43 s	0.01 s	0.03 s	0.51 s	2.41 s	3 s
HiPAcc	37.08 s	0.01 s	0.05 s	0.6 s	5.5 s	3 s
Trimesh	180.98 s	0.01 s	0.05 s	0.71 s	5.57 s	3.1 s
MOTIV	Time-out	0.01 s	Time-out	3.15 s	Time-out	4.61 s
WeaFQAs	Time-out	0.01 s	Time-out	2.9 s	Time-out	3.68 s
Fiasco	Time-out	0.01 s	Time-out	1.11 s	Time-out	3.2 s
axTLS	Time-out	0.01 s	Time-out	2.45 s	Time-out	8.13 s
uClibc-ng	Time-out	0.01 s	Time-out	4.95 s	Time-out	9.73 s
Busybox 1.1	Time-out	4.3 h	Time-out	Time-out	Time-out	Time-out
Busybox 1.2	Time-out	5 h	Time-out	Time-out	Time-out	Time-out
Linux 2.6	Time-out	Time-out	Time-out	Time-out	Time-out	Time-out

Munoz et al. – The Journal of Systems and Software 204 (2023) 111770

With $Nemo_2$, classical solvers can count and sample real-world NFMs configuration spaces of sizes of $< 10^{248}$ and $< 10^{45}$ respectively

RQ1. Results and Answers

Near-Optimal: **U**niform **R**andom **S**ampling **(URS)** / **S**tatistical **R**ecursive **S**earch **(SRS)** + **M**ann-**W**hitney **U** (MWU)

NFM	*N*	*rSRS*	*rURS*	*rTest*	*pSRS*	*pURS*	*pTest*	*rBetter*
Dune	71.32	0.007	0.016	Pass	0.039	0.042	Pass	93%
HSMGP	66.42	0.008	0.017	Pass	0.005	0.011	Pass	91%
HiPAcc	65.82	0.010	0.017	Pass	0.002	0.004	Pass	82%
Trimesh	129.21	0.003	0.009	Pass	0.003	0.013	Pass	91%

Finding Near-Optimal Configurations for FSE2015 Systems

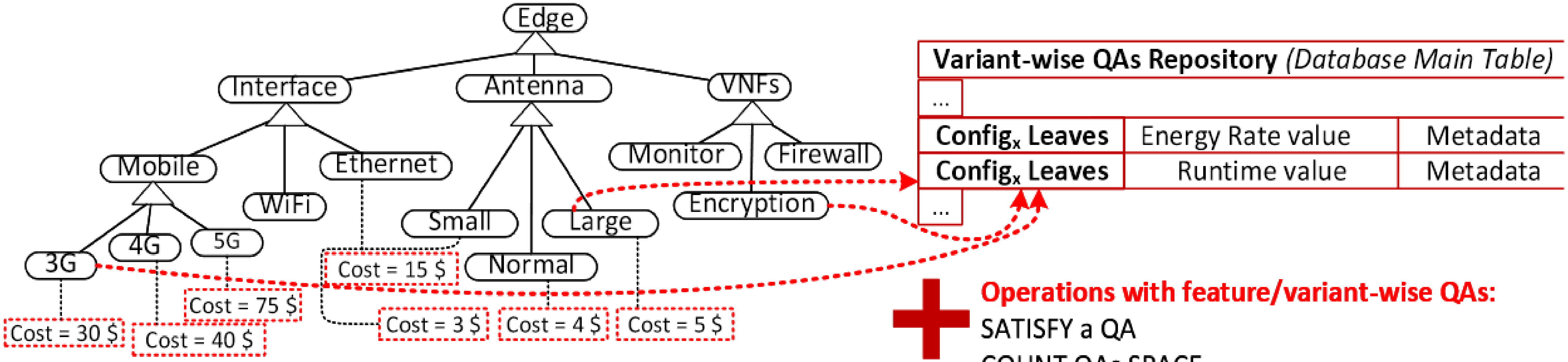
Munoz et al. – Proceedings of the 23th ACM International Systems and Software Product Line Conference

With $Nemo_2$ and URS, we find optimals near 1.4% of the absolute optimal. With SRS we assure an optimal configuration near the 0.8%.

RQ2 Approach
Eco-Assistant Hybrid Reasoning

RQ2 Main Focus: Variant-wise QAs

- **Variant-wise**: Valued QAs of sets of features that form complete configurations.
- The QA values are too heterogeneous to define an accurate aggregation formula.
- Consequently, they cannot be modelled in a extended NFM.



Operations with feature/variant-wise QAs:
SATISFY a QA
COUNT QAs SPACE
FILTER by/with QAs
CONFIGURATION WITH QA VALUES GENERATION
...

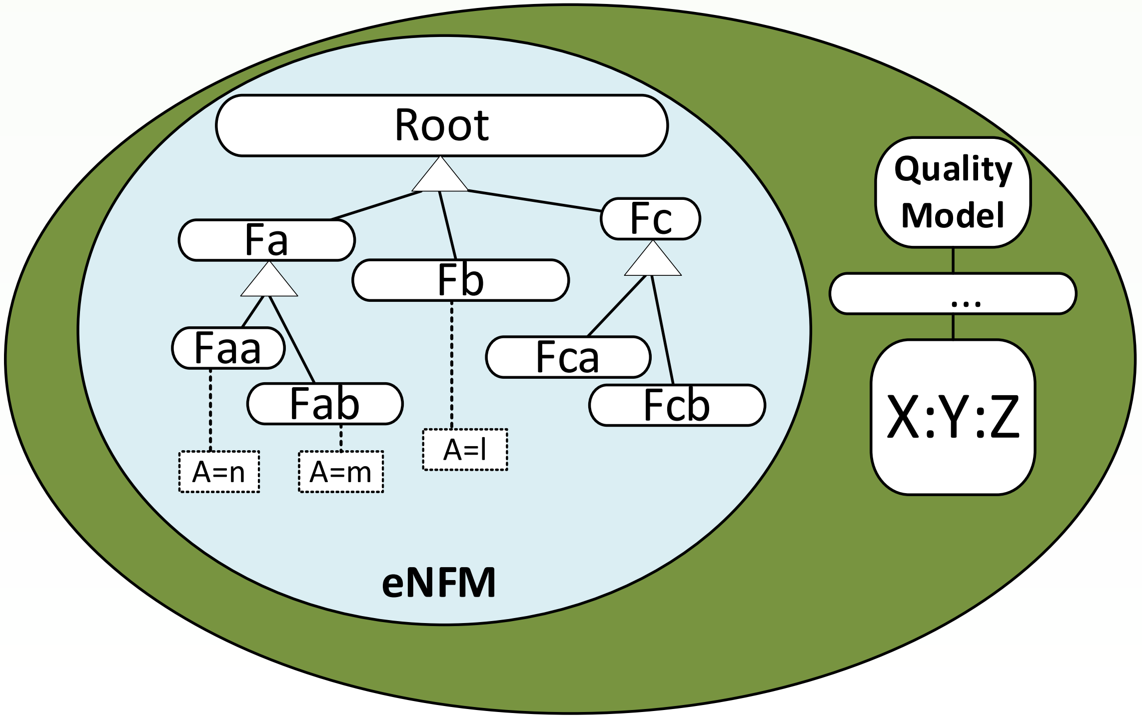
Operations without QAs:
SATISFY, COUNT, FILTER,
CONFIGURATION GENERATION
...

cross-tree constaints: Ethernet excludes Antenna

Munoz et al. – Proceedings of the 26th ACM International Systems and Software Product Line Conference

RQ2: Contributions 1 and 2

1. Define a database schema to store QMs with valued **variant-wise QAs** in **collaborative** relational **database.**
2. Develop a **PHP algorithm** that soft-links extended NFMs and Energy values by indexing configuration leaves IDs. Technically: Clafer↔PHP Query↔MariaDB.



Feature

ID	Name	Cardinality	Parent
1	Root	XOR	1*
2	Fa	XOR	1*
3	Faa	XOR	2*
4	Fab	XOR	2*
5	Fb	XOR	1*
6	Fc	XOR	1*
7	Fca	XOR	6*
8	Fcb	XOR	6*

Quality Model

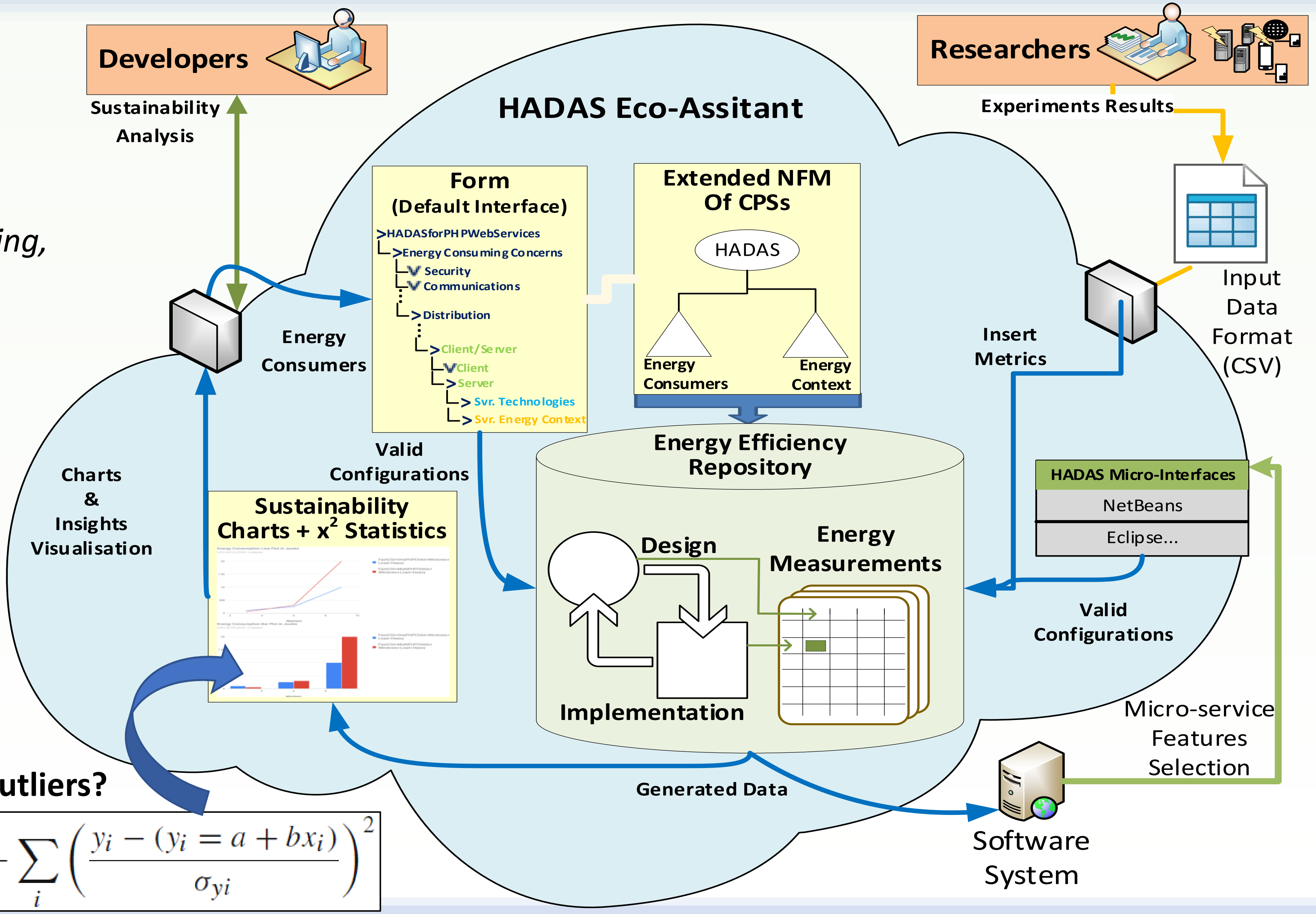
ID	Name	Domain	Parent
-	X	Z	-*
...			

Feature Attribute

ID	Name	Value	Feature
1	A	n	3*
2	A	m	4*
3	A	l	5*

RQ2 Tool Support: *HADAS* - https://hadas.caosd.lcc.uma.es/

Munoz et al. – Computing, 100:1155–1173, 2018



x_i ≡ Feature i

y_i ≡ QA

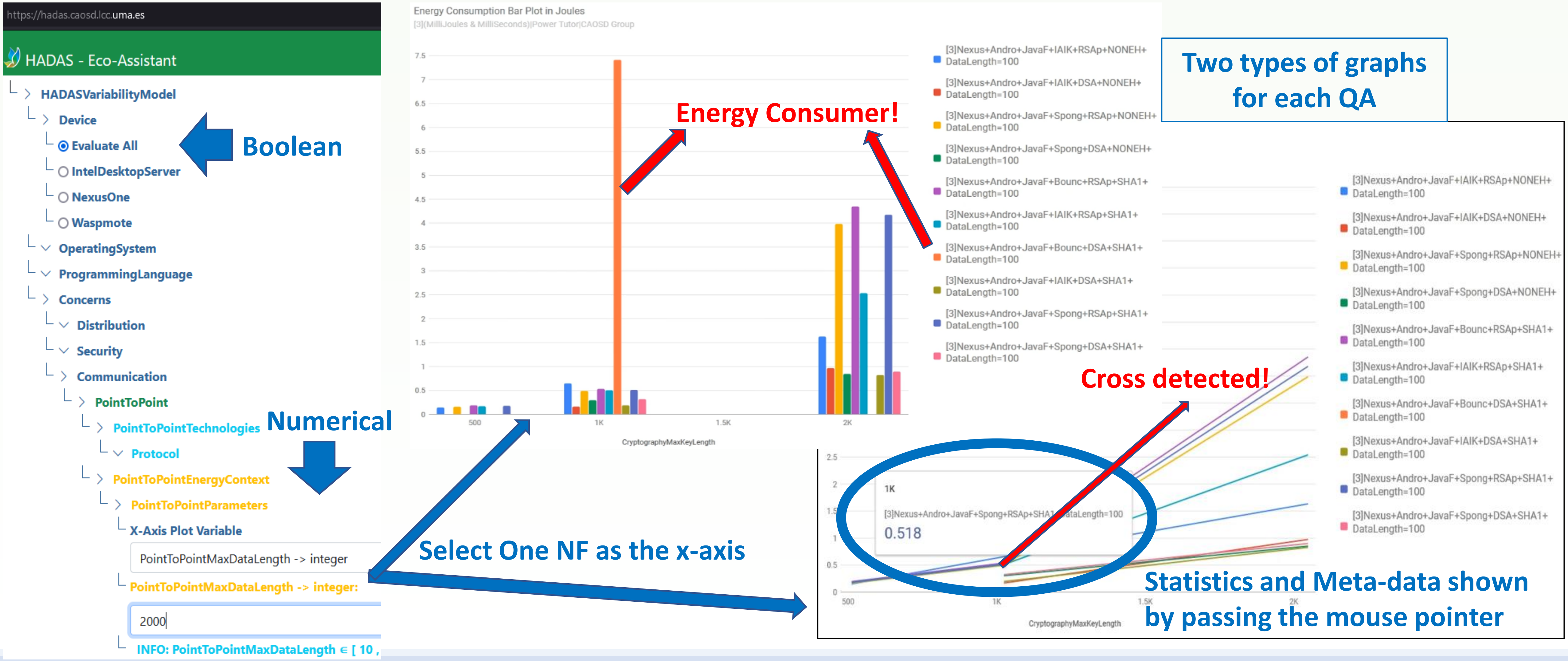
σ_{yi} ≡ Estimated error

Outliers?

$$\Delta\tilde{\chi}^2 = \sum_i \left(\frac{y_i - \overline{y}}{\sigma_{yi}}\right)^2 - \sum_i \left(\frac{y_i - (y_i = a + bx_i)}{\sigma_{yi}}\right)^2$$

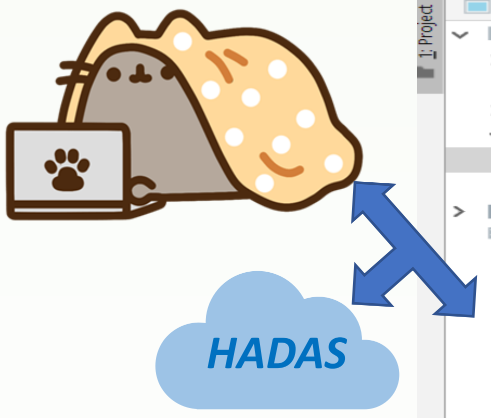
RQ2: Contribution 3

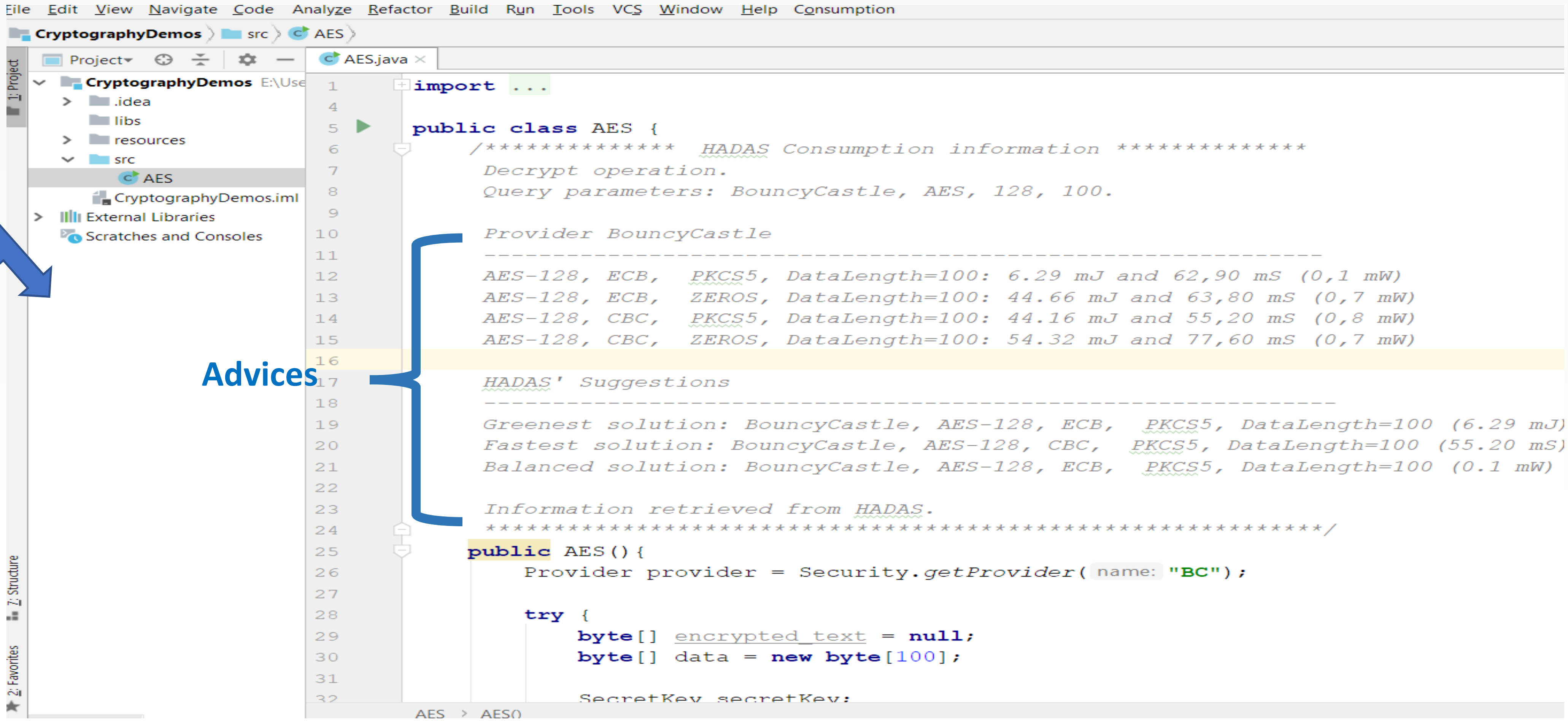
Interactive user interface for energy and runtime **statistical** reasoning



RQ2: Contribution 4

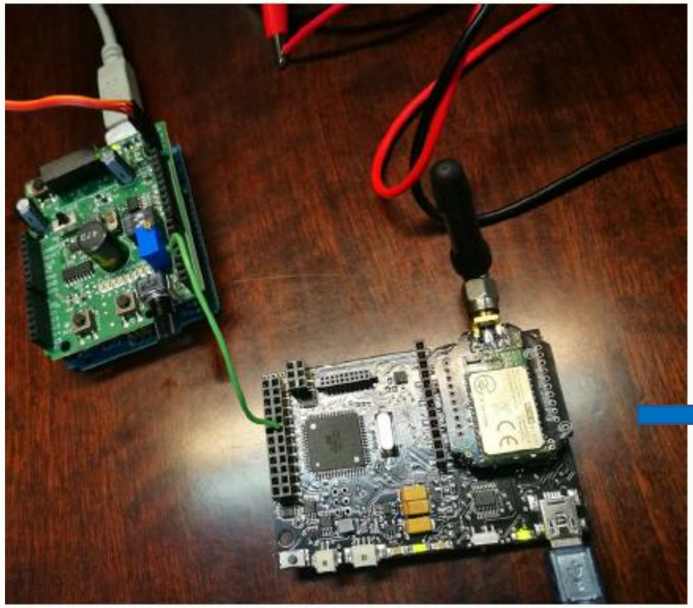
IDE Plugin that **dynamically** parses the code as features, and connects to HADAS microservice to suggest **energy/time/trade-off** efficient alternatives

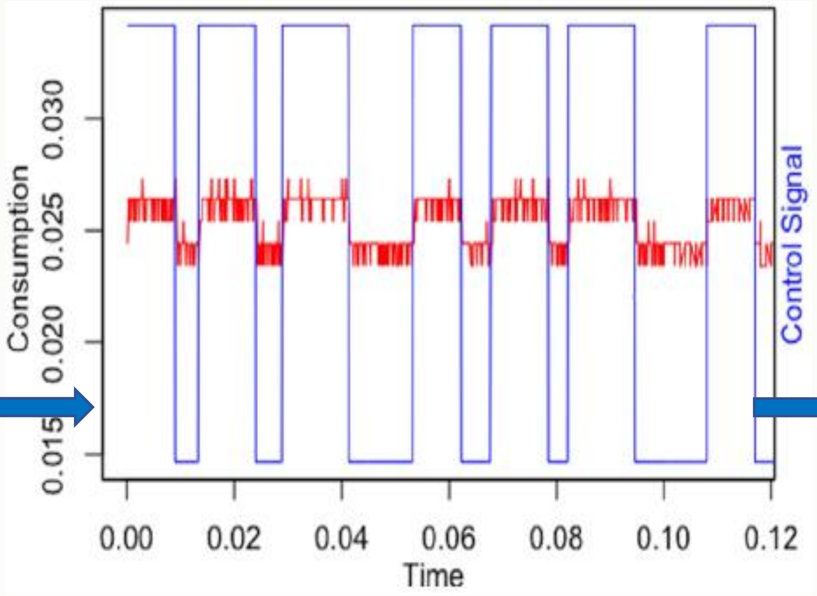




Munoz et al. – Future Generations in Computer Systems, vol 91, 2019

RQ2: Third-party and Own Energy Measurements





Waspmote Communication consumption.

	Bluetooth			WiFi		
	Software	Hardware		Software	Hardware	
Data length	Time	Time	Cons.	Time	Time	Cons.
10	100.06	99,42	3,38	3,56	3,10	0,19
100	107,98	107,26	3,62	11,56	10,71	0,65
1000	187,00	186,41	9,65	89,00	89,74	5,75
2000	273,29	273,22	13,28	176,40	176,27	11,43

Munoz et al. – Future Generations in Computer Systems, vol 91, 2019

- Green-Miner, Green Scaler and Green Oracle energy measurements CSV of Java collection classes and Android applications.
- PHP benchmarks provided by the Phoronix Test Suite.
- Low level C-compiler energy readings with the MAGEEC measurement board from the University of Bristol – *provided in a research stay in the micro-controllers group*.

RQ2: Evaluation with a Survey to Experienced Developers

- 17 participants with different expertise evaluated *HADAS* with a Likert scale.

Likert-scale question	Average
1. The developers have considered a similar solution before	0.4
2. The plugin is user-friendly	3.6
3. The information provided by the plugin is understandable	3.9
4. The plugin interface is consistent with the IDE	3.9
5. The prompt to constraint the variability is clear	3.9
6. The plugin is complex	4.1
7. The developers will recommend the plugin	3.8
8. The developers will make use of the plugin	3.8

Munoz et al. – Future Generations in Computer Systems, vol 91, 2019

Professional programmers recommend *HADAS* usage 3.9/5

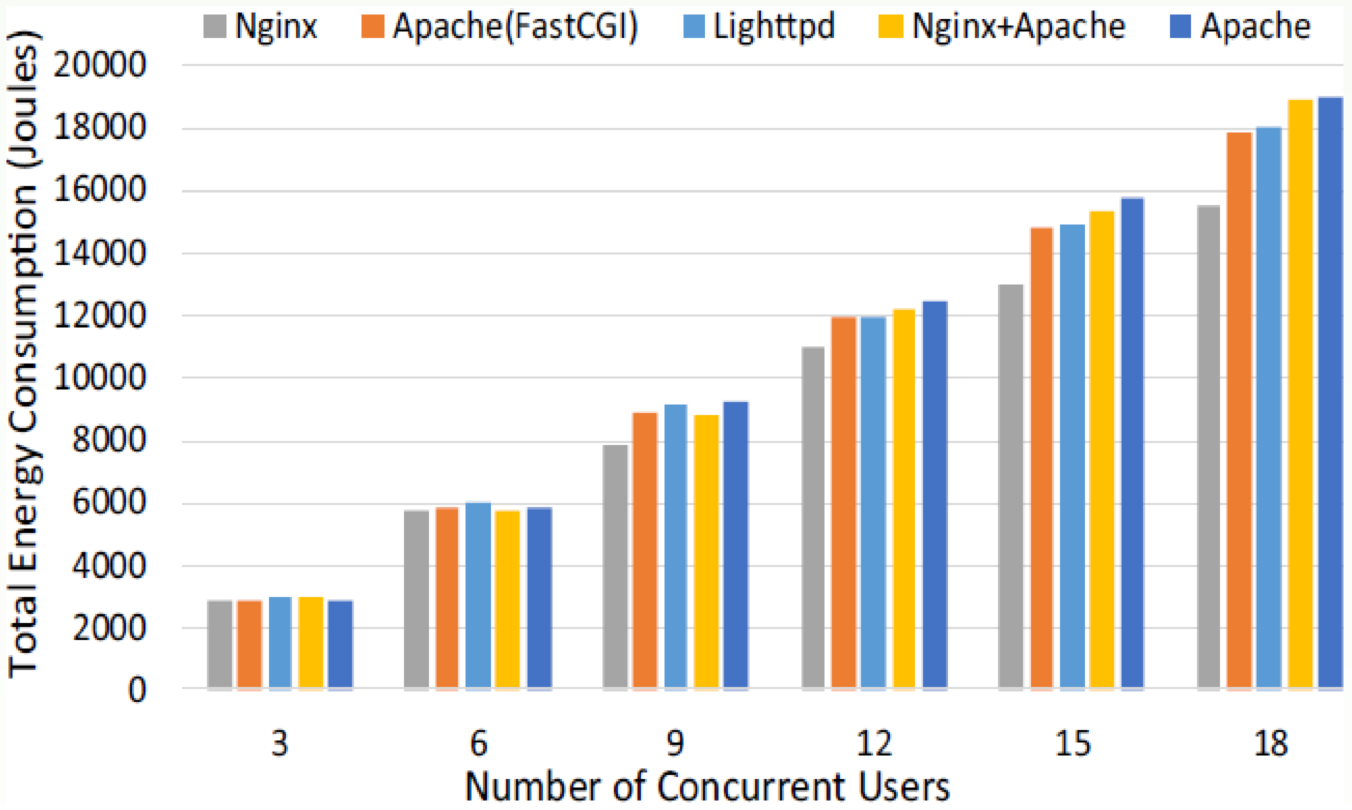
RQ2: Benchmarking *HADAS* Scalability

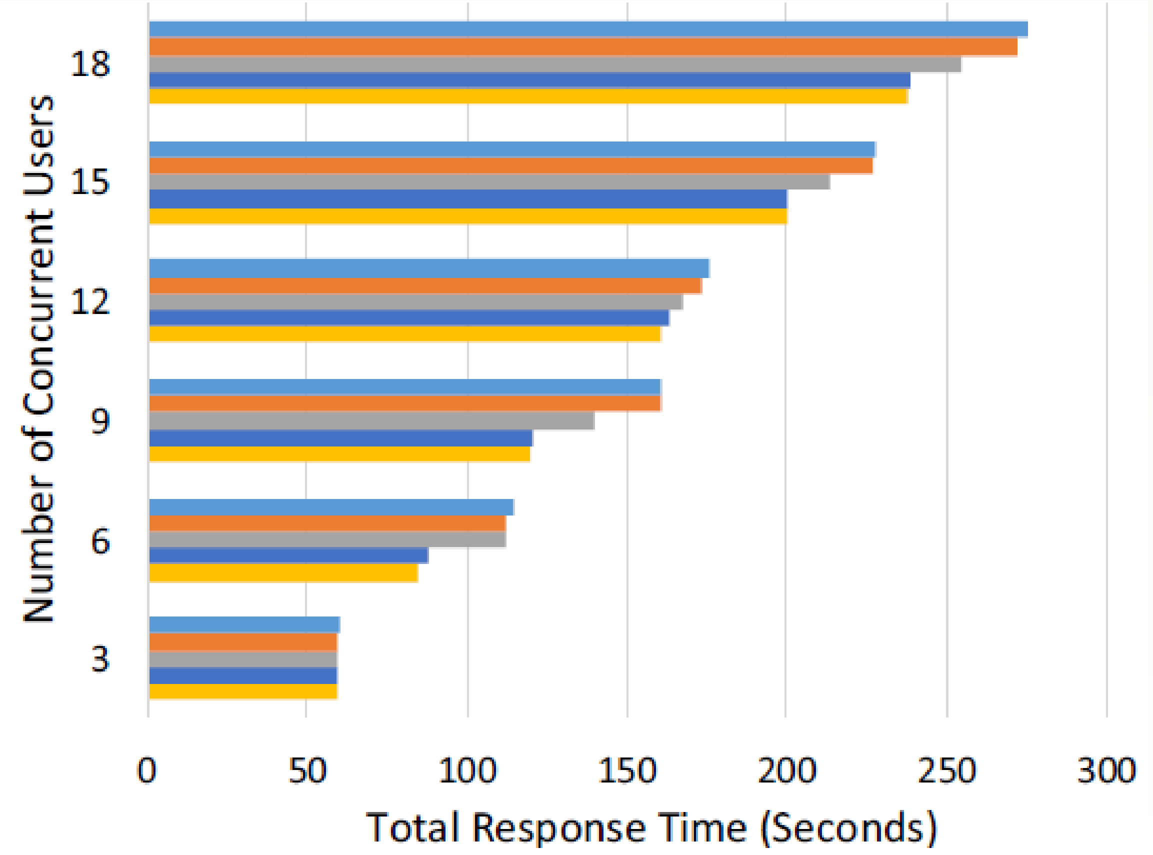
Configuration Size	# Configurations	Milliseconds
5	**10**	**4**
5	50	6
10	**10**	**10**
10	200	16
20	**10**	**32**
20	800	54

Munoz et al. – Tool Proceedings of the 23th ACM International Systems and Software Product Line Conference

HADAS **processing is in the ms region for common CPSs analyses**
The number of features affects more than the configuration space

RQ2: *HADAS* Results and Answers





Munoz et al. – Computing, 100:1155–1173, 2018

***HADAS* reduced the <u>energy consumption</u> of CPSs up to 70% and of web servers running PHP services up to 25%**

RQ2: Partially Measured Colossal Configuration Spaces

Who wants to manually measure the energy consumption of 10^{200} *different configurations?*

What will happen if the Configuration Space is Partially Measured?

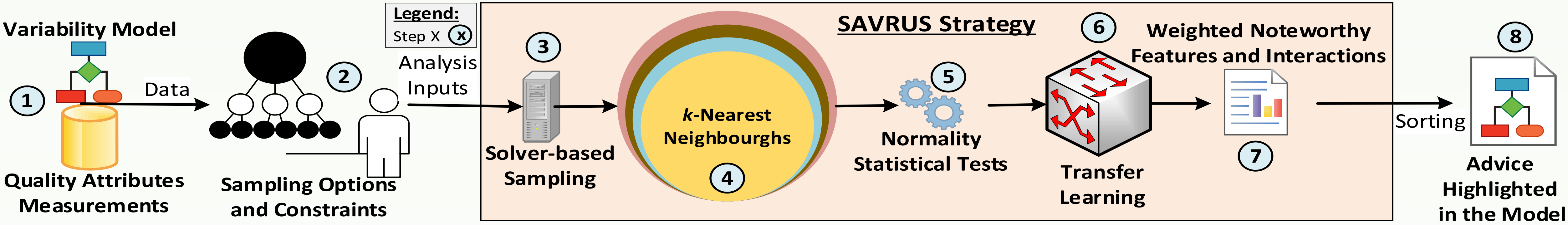
- **Measure** the energy consumption of large spaces is **unfeasible**.
- **Automatise** the measuring of heterogenous systems is **unfeasible**.
- Build **energy models** is computationally complex and they are **inaccurate**.
- We need some **advices** and **in short times** – developers are not patience.

What can we do?

RQ2: Unfortunately….

It is unsolvable and here finish this RQ

RQ2 Contribution 5: *SAVRUS* - https://hadas.caosd.lcc.uma.es/savrus.php

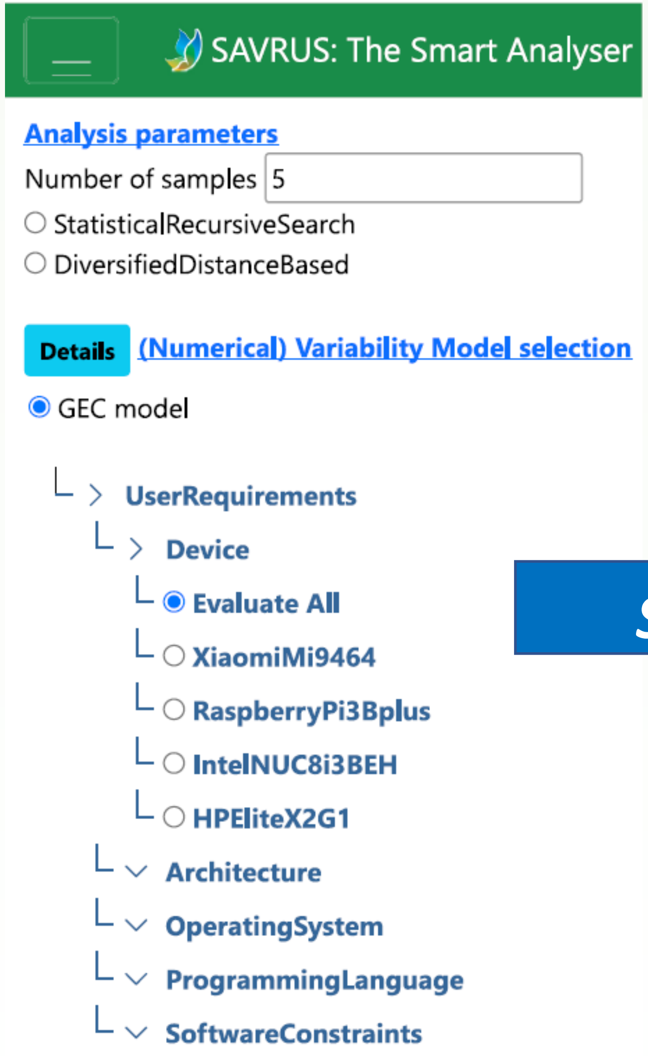


Munoz et al. – Knowledge-Based Systems 270 (2023) 110558

Modular algorithm based in **machine and transfer learning**:

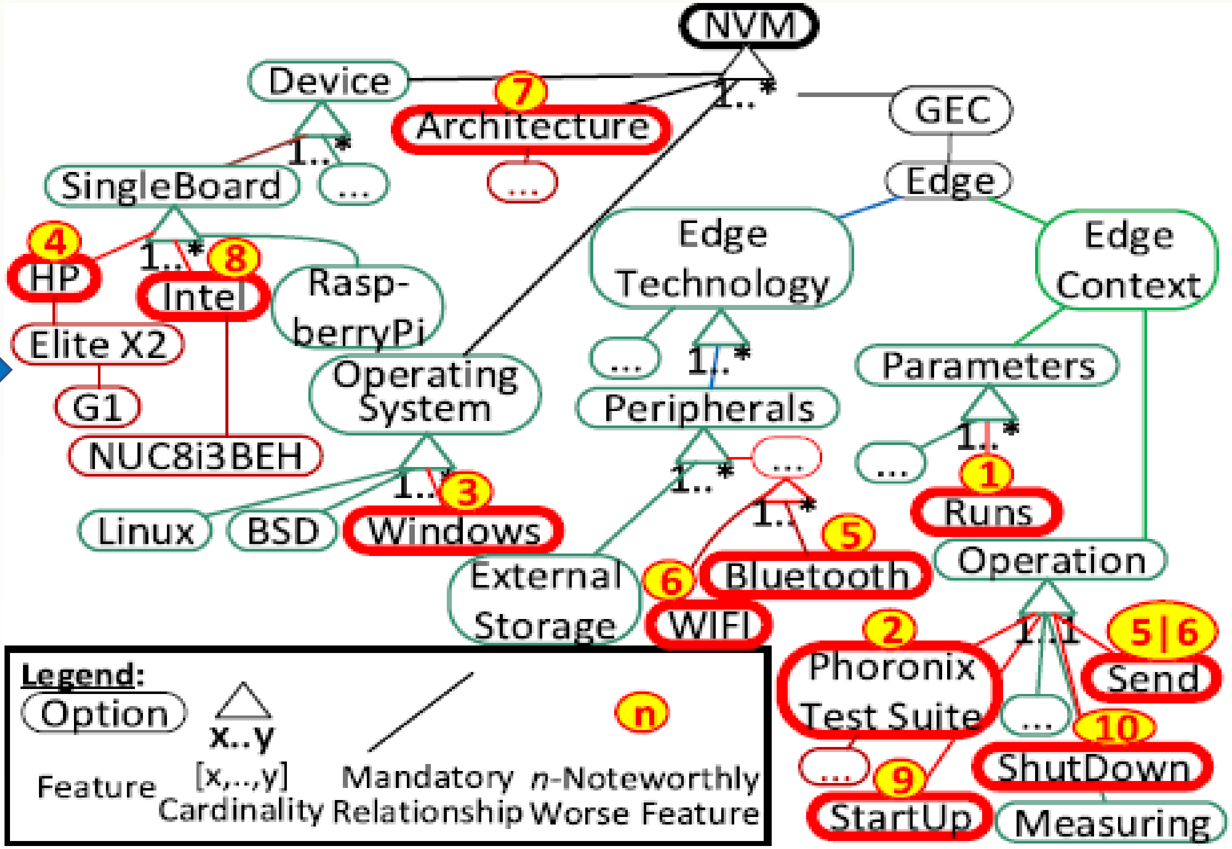
- Re-work *HADAS* for partially measured spaces.
- Step 3 **Sampling**: **SRS** or Diversified Distance-Based Sampling (**DDbS**) to generate a **subspace**.
- Step 4 Regression **approximation**:
  - ***k*NN** to approximate unmeasured samples.
  - N-dimensions Manhattan distance: $|X_1 - Y_1| + |X_2 - Y_2| + \ldots + |X_n - Y_n|$
- Step 5 **Statistic**al noteworthiness test:
  - Detect interactive features to the energy consumption up to **3-Wise***.
  - **MWU** to assured a **95% of interaction confidence**.
- Step 6 **Transfer learning*** from previous *SAVRUS* executions: **Scores** of Local Outlier Factor [0,1].

RQ2: Contribution 5 and SAVRUS Tool Result



3-wise* features ranked by their effect to energy consumption (or another QA)

SAVRUS Algorithm



Munoz et al. – Knowledge-Based Systems 270 (2023) 110558

RQ2: Real-World Models Evaluated in *SAVRUS*

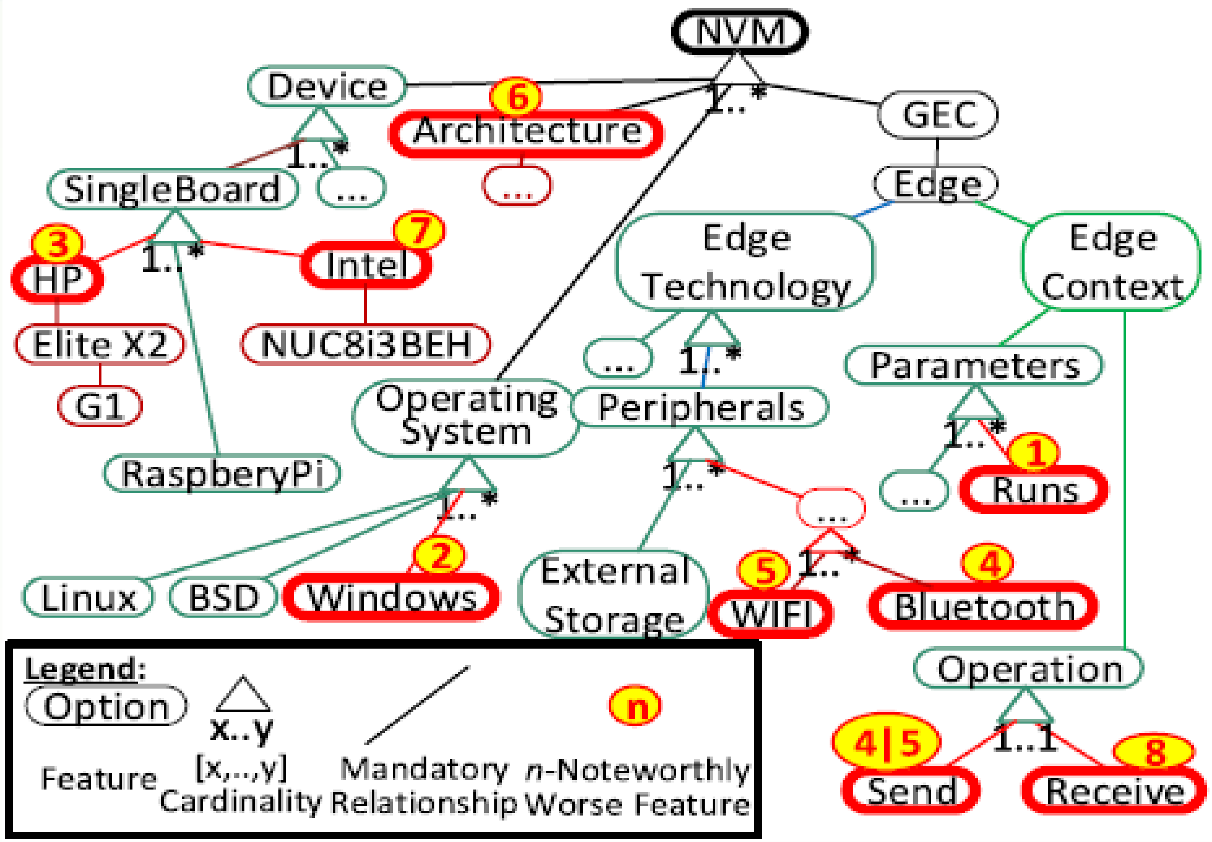
NVM	Description	#Booleans	#Numericals	Space	#Measurements
Dune	Multi-grid solver	11	3	2,304	2,304
HSMGP	Stencil-grid solver	14	3	3,456	3,456
HiPAcc	Image analysis framework	33	2	13,485	13,485
Trimesh	Triangle mesh library	13	4	239,360	239,360
GEC	Generic edge computing	**552**	2	**$\sim 5.3 * 10^8$**	132,500

Munoz et al. – Knowledge-Based Systems 270 (2023) 110558

- We need a large and partially measured space to test the results of *SAVRUS*.

- We need **completely measured spaces** to test coverage and accuracy of *SAVRUS*. In each execution we **degrade** differently their measured space.

RQ2: *SAVRUS* Results and Answers



Munoz et al. – Knowledge-Based Systems 270 (2023) 110558

GEC up to 90% energy consumption reduction by performing a sequence of SAVRUS analyses with just 3% samples of the 0.25% measurements of the total space

RQ2: *SAVRUS* Results and Answers

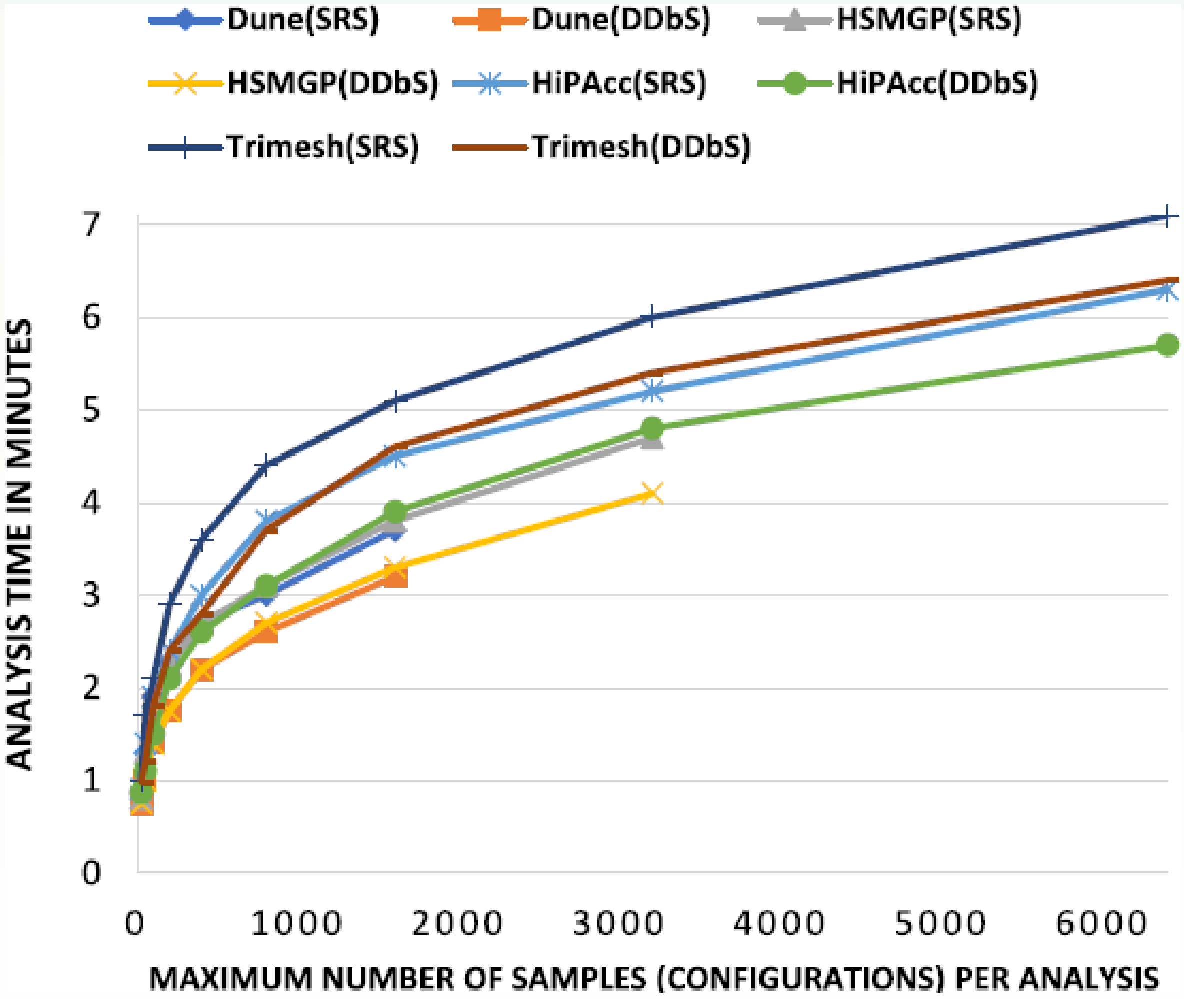
SAVRUS time, coverage and accuracy for SRS/DDbS querying 3% samples of the incompletely measured space of four real-world systems.

Sampling	SRS			DDbS		
Model						
	Time	Coverage	Accuracy	Time	Coverage	Accuracy
Dune	1.5 m	73.2%	86.7%	1.3 m	76.9%	83.9%
HSMGP	1.9 m	72.4%	85.4%	1.4 m	75.7%	83.5%
HiPAcc	3.7 m	70.5%	84.1%	2.8 m	73%	83.3%
Trimesh	7.3 m	65.2%	83%	6.6 m	68.7%	82.8%
Mean:	**3.6 m**	**70.3%**	**84.8%**	**3 m**	**73.6%**	**83.4%**

Munoz et al. – Knowledge-Based Systems 270 (2023) 110558

- **SAVRUS generates a correct ranking of noteworthy features and interactions in average ~80% of the times with a similar coverage**
- **SRS is more accurate but slower than DDbS**

RQ2: Benchmarking *SAVRUS* Scalability



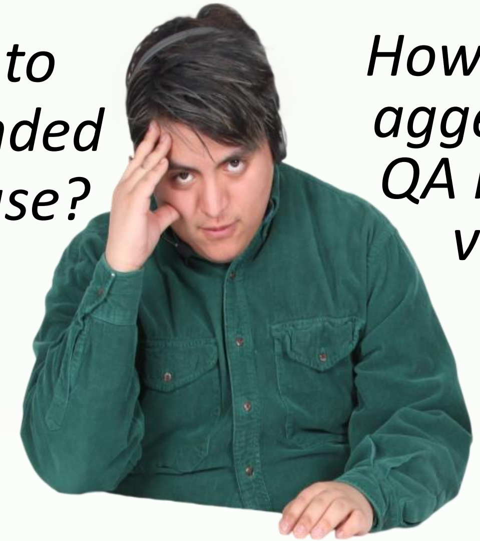
Munoz et al. – Knowledge-Based Systems 270 (2023) 110558

SAVRUS has base runtime of ~1 minute, taking less than 3 minutes for comprehensible executions and ~7 minutes in the worst cases.


RQ3

RQ3: Unify Variability and Quality Modelling

Do I really need to maintain an extended FM and a Database?



How can I define that an aggegated feature-wise QA is interacting with a variant-wise one?

- CPSs require **quality-aware** and advance reasoning:
  - Classical operations: SAT, compute configurations, model counting, etc.
  - <u>Quality-aware operations</u>: QAs SAT, generate/count measured spaces, etc.
  - <u>Interactions between QAs</u>: Constraints between Feature-wise and Variant-wise QA values.
  - <u>Optimisation</u>: Multi-objective, valued trade-offs, etc.

Formally define a unified extended NFM and valued QM standard

RQ3 BACKGROUND

Why Are You Using the Exotic Category Theory? – Reviewer X

Just focus on the reddish boxes

→Model ↓*Entity*	NVM	Category Theory	Set Theory	FODA	Codd Algebra	HOL	Arith-metic
Structured Model	Labelled NVM	Category	Finite Set	Labelled FM	Data Schema	Logic Formula	Plane
Entities	Sub-tree, Feature, Solution	Sub-Category, Object, Instance	Sub-Set, Element	Sub-tree, Feature, Configu-ration	*Table ,Cell*	Partial Formula, Variable	Sub-System, Equation ,Variable
Boolean Type	$\mathbb{B}$ Feature	$\mathbb{B}$	$\mathbb{B}$	Feature	$\mathbb{B}$	$\mathbb{B}$	*Pseudo-$\mathbb{B}$: [0,1]*
Numerical Type	$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$ Feature	$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$	$\mathbb{N}$, $\mathbb{Z}$ *Finite Sets*	*Un-supported*	$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$	*Un-supported*	$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$
Selection	Assert	Δ	$\in$	Require	$\pi_{[\text{Column}]}$	True	$=$
Exclusion	Not	$-\Delta$	$\not\subseteq$	Exclude	$-\pi_{[\text{Column}]}$	False	$\neq$
Con-nectives	&&, |||, [x..y], $\Rightarrow$, $\equiv$	$\sum_{[\text{Functor}]}$ $\prod_{[\text{Functor}]}$, X	$\wedge$, $\vee$, $\oplus$, If, $\Rightarrow$, $\equiv$	And, Or, xOR, $\Rightarrow$, $\equiv$	Foreign Key, $\bigcap$, $\bigcup$, $\biguplus$, Joins$_{[X]}$	$\wedge$, $\vee$, $\oplus$, If, $\Rightarrow$, $\equiv$ $\forall$, $\exists$, !	Equation Systems
Equalities	$=, \neq, >, \geq, <, \leq$	$=, \neq, >, \geq, <, \leq$	$=, \neq$	$=, \neq$	$=, \neq, >, \geq, <, \leq$	$=, \neq$	$=, \neq, >, \geq, <, \leq$
Mathe-matics	+, -, *, ÷, %...	+, -, *, ÷, %...	*Pre/Suc-cessor*	*Un-supported*	*Un-supported*	*Un-supported*	+, -, *, ÷, %...

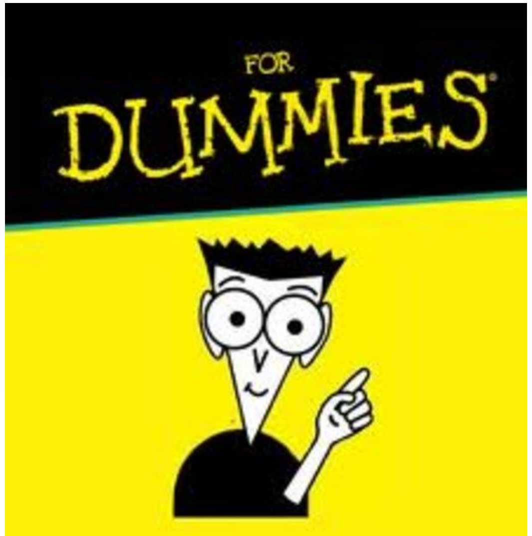
Category Theory supports everything natively

Munoz et al. – Proceedings of the 33rd International Conference on Advanced Information Systems Engineering CAISE

Fundamentals of Category Theory

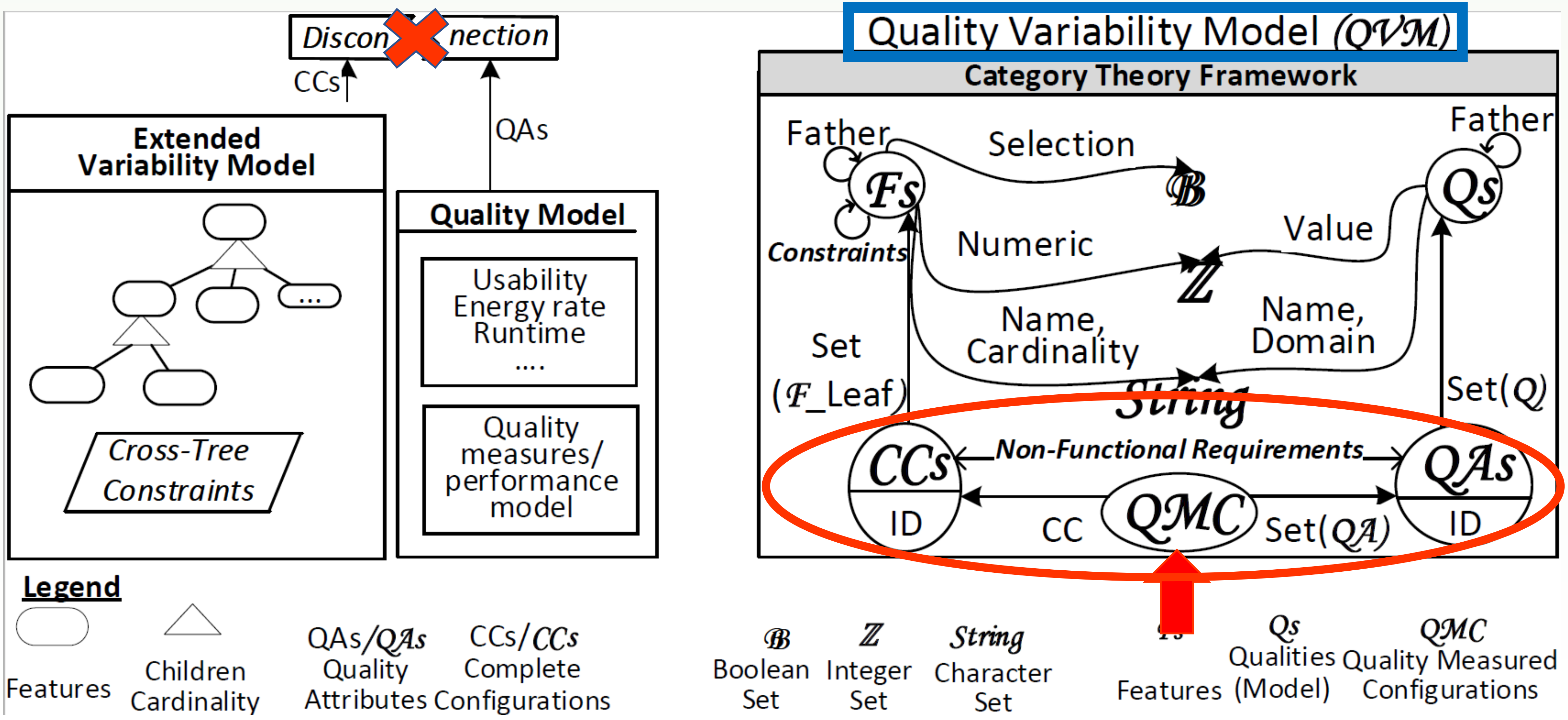
Algebraic theory of mathematical structures to capture and relate similar aspects of structures while abstracting from the individual specifics.

- **Category**: A set of <u>Objects</u> and <u>Arrows</u> in a labelled directed graph.
- **Objects**: A structured template $X \in Ob(C)$, graphically depicted as a node $\bullet^{X}$.
- **Arrows**: A structure-preserving function $a \in Arr(C)$ with source and target objects and identity and composition properties depicted $\bullet^{X} \rightarrow^{a} \bullet^{Y}$.
- (Generalised) **Element**: Arrows to data domains.
  - E.g., $\bullet^{X_{e_1}} \rightarrow^{Name}$ String
- **Functor**: A process F between categories depicted $\bullet^{c_1} \rightarrow^{F} \bullet^{c_2}$.
- **Instance**: Set-valued functor that assigns values to elements.
  - E.g., $\bullet^{X_{e_1}} \rightarrow^{Name}$ "Java", and $\bullet^{X_{e_2}} \rightarrow^{Value}$ True



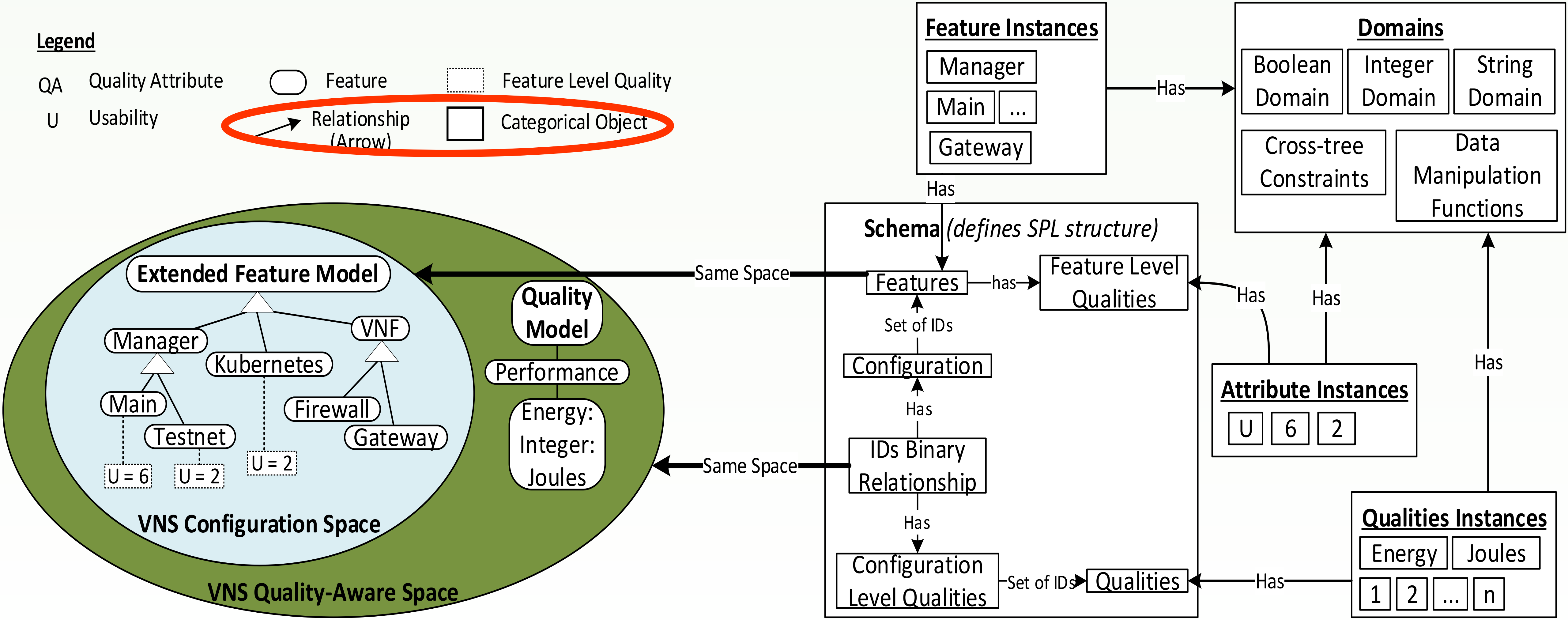
RQ3 Approach
Quality-Aware Unified Reasoning

RQ3: Contribution 1 – The Categorical QVM



Munoz et al. – Proceedings of the 33rd International Conference on Advanced Information Systems Engineering CAISE

RQ3: Instantiating a *QVM*



RQ3: Modelling a CPS $\mathcal{QVM}$ in CQL IDE

schema MyVMCategory = literal : Ty {..1..2..3..4..5..}
Instance CPSQVM = literal : QVMCategory {
# Equations defining the CPS extended NFM
# Equations for the 63 measured configurations of the CPS
# Equations defining QAs values and its metrics of CPS
# Equations connecting configurations to QA values}

1 **#Extended Numerical Variability Model Arrows**

feature : VM -> String # Feature Name

value : VM -> Integer # Numerical Feature Value

domain : VM -> String # Feature Domain (Boolean by default)

cardinality : VM -> String # Children cardinality ([x..y])

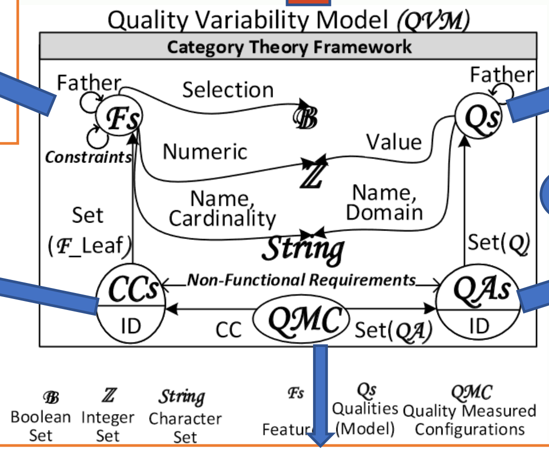
cost_USD : VM -> Integer # Feature-wise Quality Attribute Cost

2 **# Quality Model Arrows**

name : Qn -> String # Quality Name

value : Qv -> Integer # Quality Numerical Value

domain : Qd -> String # Quality Domain



3 **# IDs for QAs relationship**

id : QAs -> Integer #Identify each set of Valued Qs

4 **# IDs for configuration relationship**

id : CCs -> Integer #Identify each Complete Configuration

5 **# QMC SPAN (binary relation between CCs and sets of QAs)**

phi : QMC -> Integer # "Point" to a Complete Configuration

psi : QMC -> Integer # "Point" to a set of Valued Quality Attributes

Munoz et al. – Proceedings of the 26th Software Product Line Conference SPLC

RQ3: Contribution 2|The QVM Framework - https://www.categoricaldata.net/

Algorithm 1: Categorical Model Report

Input: Populated $\mathcal{F}s$, CCs, QMC, $Q\mathcal{A}s$

(bool, num, logic, arithm, confs, measured[]) = [0, . . ., 0];

bool = **Add**($\lambda(x \in \mathcal{F}s \mid !x.num) : 1$);

num = **Add**($\lambda(x \in \mathcal{F}s \mid x.num) : 1$);

logic = **Add**($\lambda(x \in \mathcal{F}s.where \mid x.ops \notin [=, +, ..., \%]) : 1$);

arithm = **Add**($\lambda(x \in \mathcal{F}s.where \mid x.ops \in [=, +, ..., \%]) : 1$);

confs = **Add**($\lambda(CCs.id) : 1$);

measured[QA]=**Add**($\lambda(CC.f.att \vee CC.id \in QMC) : 1$);

(FQAs, CCQAs) = [' ', ' '];

FQAs = **ConcatIfNew**($\lambda(x \in \mathcal{F}s.att)$:
$[x.name, x.val, x.dom, x.AggregationFunction]$);

CCQAs =
ConcatIfNew($\lambda(x \in QAs) : [x.name, x.val, x.dom]$);

Result: bool, num, logic, arithm, confs, measured, FQAS, CCQAs

Optimal, as we only query the required objects

Munoz et al. – Proceedings of the 26th Software Product Line Conference SPLC

Algorithm 2: Categorical Aggregation of Attributes

Input: Populated CCs

AggregatedQAs = [][];

forall $cc \in CCs$ **do**

Func = [];

Func = **Push**($\lambda(x \in cc.feature.att) : x.function$);

AggregatedQAs[cc.id] = **Push**(
$\lambda(x \in cc.feature.att, f \in Func \mid f = x.function) : f(x)$);

end

Result: AggregatedQAs

Algorithm 3: Categorical Quality Optimisation

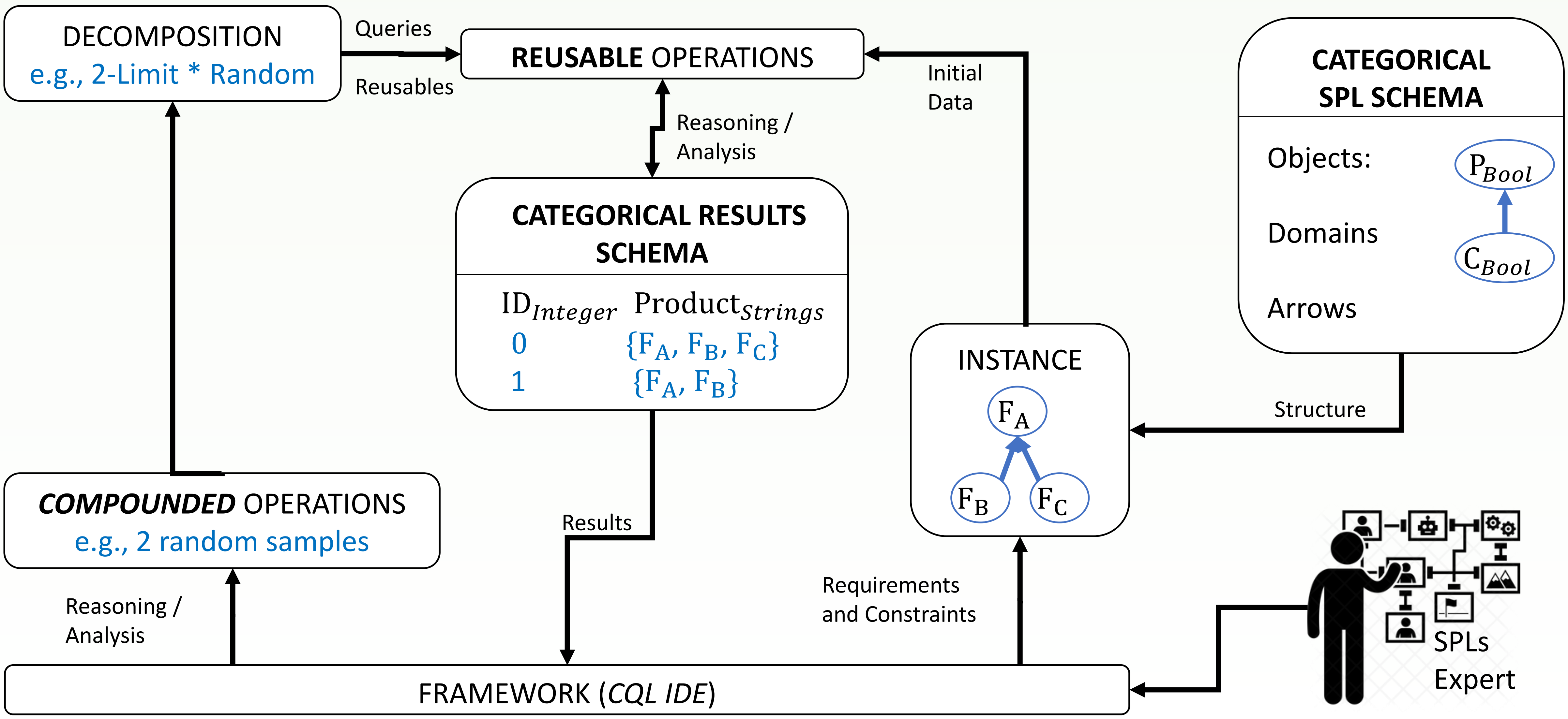
Input: Populated QMC, CCs, Objectives

eQMC = [QMC.cc, (QMC.qa ∪ Aggregation(CCs))];

Result: $(\lambda x \in eQMC \mid Objectives(x.qas)$:
$[x.cc.features, x.qas])$

Formal and algorithmic definition of the QVM Quality-Aware Operations

RQ3: Usage of the QVM Framework - URS



RQ3: CPS Filter Reasoning in CQL IDE

CAOSD

We implemented in CQL IDE the Quality-Aware Reasoning Operations

Energy=<18 & Runtime =<10 (36 out of 63 configurations)

pizza.cql - 16:31:04

- Summary
- typeside Ty
- schema PizzaCategory
- instance PizzaData : Pi
- schema RESSATISFY
- query SATISFY : PizzaC
- instance SATSolver : RI
- schema RESCOUNT
- query COUNT : PizzaCa
- instance COUNTSolver
- schema RES
- query FILTER : PizzaCat
- instance FILTERSolver :
- query BOUND : PizzaC
- instance BOUNDSolver
- query RANDOM : Pizza
- instance PizzaDataRan
- instance RANDOMSolv
- schema RESAGGREGAT
- query AGGREGATE : Pi
- instance AGGREGATES
- query FILTERAGGREGA
- instance FILTERAGGRE

Tables | TyAlg | Hom-sets | DP | Graph | Text | Expression

CONFS (36)

QA FILTER Reasoning

..	cc_traces	qas
0	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 14 Watts & Runtime = 2 Seconds
1	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 13 Watts & Runtime = 2 Seconds
2	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 13 Watts & Runtime = 2 Seconds
3	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 13 Watts & Runtime = 2 Seconds
4	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 14 Watts & Runtime = 2 Seconds
5	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 13 Watts & Runtime = 2 Seconds
6	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 13 Watts & Runtime = 2 Seconds
7	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 13 Watts & Runtime = 2 Seconds
8	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 14 Watts & Runtime = 1 Seconds
9	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 13 Watts & Runtime = 1 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 13 Watts & Runtime = 1 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Monitor &...	& EnergyRate = 13 Watts & Runtime = 1 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Firewall & ..	& EnergyRate = 18 Watts & Runtime = 6 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Firewall & ..	& EnergyRate = 17 Watts & Runtime = 5 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Firewall & ..	& EnergyRate = 16 Watts & Runtime = 4 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Firewall & ..	& EnergyRate = 15 Watts & Runtime = 3 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Firewall & ..	& EnergyRate = 18 Watts & Runtime = 6 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Firewall & ..	& EnergyRate = 17 Watts & Runtime = 5 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Firewall & ..	& EnergyRate = 16 Watts & Runtime = 4 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Firewall & ..	& EnergyRate = 15 Watts & Runtime = 3 Seconds
...	& Interface -->(XOR) VirtualNetworkFunctions -->(XOR) Firewall & ..	& EnergyRate = 18 Watts & Runtime = 6 Seconds

36 IDs, 6 nulls, 0.658 seconds. Provenance: Row limit:

RQ3: Real-world QVMs Evaluated

SPL	Description	Features	Constraints	Configurations	Quality Attributes
Pizza	Italian vendor machine [28]	• Boolean: 12 • Numerical: 1	*Empty*	• Total SAT: 42 • Measured: 84	• Feature level: (1) Cost ∈ (5,25) $: Function: Addition • Configuration level: (2) Time ∈ $(1,10^3)$ *seconds*
Truck	Truck factory [45]	• Boolean: 33	• Logic: 10	• Total SAT: 234 • Measured: 234	• Feature level: (1) Size ∈ [0,10) *metres*^{2}: Function: Product
JHipster	Software generator [24]	• Boolean: 45	• Logic: 13	• Total SAT: 26,256 • Measured: 105,024	• Feature level: (1) Usability ∈ (0, 10): Function: Addition (2) Battery ∈ (0, 20): Function: Addition (3) Footprint ∈ (0, 10): Function: Addition • Configuration level: (4) Compileable ∈ [true, false]
(1) *VNS* and (2) *FullVNS*	Virtual Network System	Figure 1 version: • Boolean: 18 • Numerical: 1 Full version: • Boolean: 40 • Numerical: 3	Figure 1 version: • Logic: 1 • Arithmetic: 1 Full version: • Logic: 63 • Arithmetic: 4	Figure 1 version: • Total SAT: 63 • Measured: 315 Full version: • Total SAT: 2,130,000 • Measured: 10,650,000	• Feature level: (1) Dependency ∈ [L, M, H]: Function: Maximum (2) Usability ∈ (1, 10): Function: Mean (3) Security ∈ [L, M, H]: Function: Minimum • Configuration level: (4) Time ∈ $(1,10^3)$ *seconds* (5) Energy ∈ $(1,10^3)$ *joules*

Munoz et al. – Proceedings of the 26th Software Product Line Conference SPLC

RQ3: Results

Reasoner:	ClaferMoo				AAFM Python Framework				CQL IDE			
SPLs:	*Pizza*	*VNS|Full*	*Truck*	*JHipster*	*Pizza*	*VNS|Full*	*Truck*	*JHipster*	*Pizza*	*VNS|Full*	*Truck*	*JHipster*
Features	0.2 s	0.2|2.7 s	0.2 s	0.3 s	0.02 s	0.02|0.39 s	0.03 s	0.05 s	0.04 s	0.05|0.55 s	0.06 s	0.07 s
Constraints	0.2 s	0.2|2.73 s	0.2 s	0.3 s	0.02 s	0.02|0.39 s	0.03 s	0.05 s	0.04 s	0.05|0.55 s	0.06 s	0.07 s
Configurations	*0.5 s	*1.05|213 s	*1.4 s	*2.7 s	*0.15 s	*0.2|60 s	*0.24 s	*5.03 s	0.17 s	0.19|156.4 s	0.22 s	1.98 s
QAs	*0.9 s	*1.7|4.3 s	*2.1 s	*2.9 s	Unsupported	Unsupported	Unsupported	Unsupported	0.06 s	0.11|3.4 s	0.14 s	0.19 s
Mean:	*0.45 s	*0.79|55.6 s	*0.98 s	*1.55 s	*0.06 s	*0.09|20.2 s	*0.1 s	*1.71 s	0.08 s	0.1|40.3 s	0.12 s	0.58 s

Reasoner:	ClaferMoo				SATIBEA				CQL IDE			
SPLs:	*Pizza*	*VNS|Full*	*Truck*	*JHipster*	*Pizza*	*VNS|Full*	*Truck*	*JHipster*	*Pizza*	*VNS|Full*	*Truck*	*JHipster*
Unconstrained	0.95 s	*1,8|236 s	2.15 s	2.99 s	1.67 s	*1.79|2.33 s	*1.85 s	1.78 s	0.27 s	0.39|211 s	0.42 s	2.68 s
Constrained	0.97 s	*1.89|237 s	2.17 s	3.01 s	Unsupported	Unsupported	Unsupported	Unsupported	0.27 s	0.4|212 s	0.42 s	2.69 s
Range	0.95 s	*1.8|232 s	2.15 s	2.94 s	Unsupported	Unsupported	Unsupported	Unsupported	0.27 s	0.41|212 s	0.43 s	2.69 s
Mean:	0.957 s	*1.83|235 s	2.16 s	2.98 s	1.67 s	*1.79|2.33 s	*1.85 s	1.78 s	0.27 s	0.4|212 s	0.423 s	2.69 s

Reasoner:	ClaferMoo				SATIBEA				CQL IDE			
SPLs:	*Pizza*	*VNS|Full*	*Truck*	*JHipster*	*Pizza*	*VNS|Full*	*Truck*	*JHipster*	*Pizza*	*VNS|Full*	*Truck*	*JHipster*
Min/Maximise	*0.98 s	*1.87|344 s	2.23 s	*4.36 s	*1.67 s	*1.79|2 s	1.85 s	*1.78 s	0.38 s	0.52|274 s	0.68 s	3.48 s
Multiobjective	*1.02 s	*1.97|355 s	2.38 s	*4.5 s	*1.85 s	*2.01|2,1 s	2.03 s	*1.95 s	0.42 s	0.67|312 s	0.78 s	3.95 s
Weighted	Unsupported	Unsupported	Unsupported	Unsupported	Unsupported	Unsupported	Unsupported	Unsupported	0.42 s	0.68|317 s	0.82 s	4.02 s
New Domain	Unsupported	Unsupported	Unsupported	Unsupported	Unsupported	Unsupported	Unsupported	Unsupported	0.41 s	0.63|274 s	0.73 s	3.47 s
Mean:	*1 s	*1.92|349 s	2.31 s	*4.43 s	*1.76 s	*1.9|2,05 s	1.94 s	*1.87 s	0.41 s	0.63|294 s	0.75 s	3.73 s

RQ3: Answers

- $\mathcal{QVM}$ is a category that formally integrates into a unifying model the variability and quality information, **natively** supporting any feature, QA, and constraints among them.

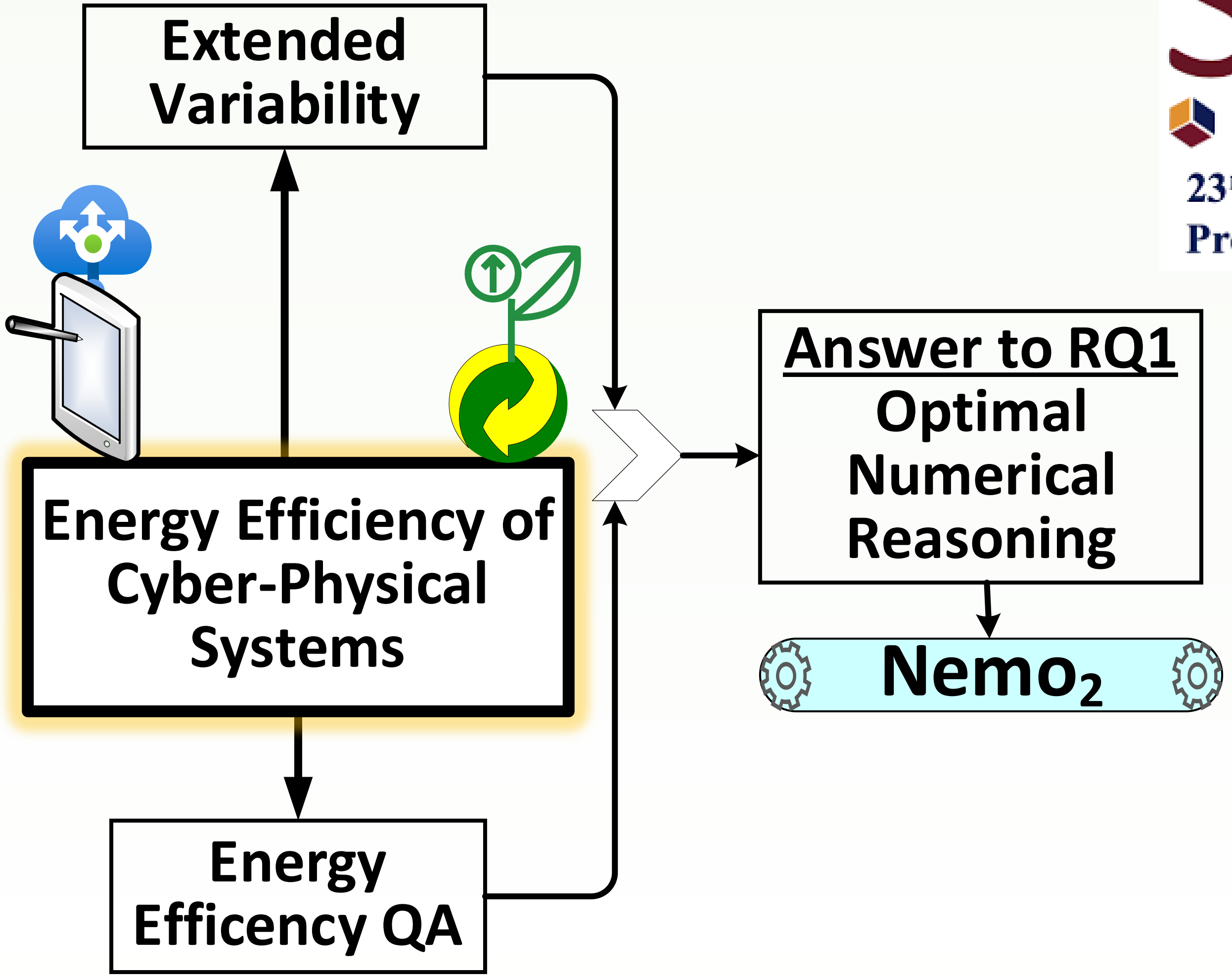
- The $\mathcal{QVM}$ framework defines QAs lambda operations for quality-aware reasoning supported by the categorical solver CQL IDE.

- The evaluation reached a total of 11 different operations, and on average, this framework was the fastest in 5 different real-world SPLs compared to popular reasoners.

Thesis Publications, Awards, Research Stays and Projects

- A total of 18 Publications:
  - 4 JCR Indexed Journals, divided in **two Deciles 1**, and **two Quartiles 2**.
  - 5 GGS Indexed Conferences, divided in **four GGS 2** and **one GGS 3**.
  - 1 *HADAS* **Tool Demonstration** paper **and poster** in a **GGS 2** Conference.
  - **2 Workshop** papers **in a GGS 2** Conferences.
  - **1 Doctoral Symposium** paper in a **GGS 2** Conference.
  - **1** International **IEEE** Conference.
  - 4 National Conferences.
- A total of 3 Awards:
  - **Best Student Paper** Award in the International Conference ES2DE 2017.
  - **Top 2 Excellence Young Scientist 2018** award of Sevilla and Málaga by the Fundación IMFAHE.
  - **Top 22 thesis in progress** of Universidad de Málaga in **2022**.
- 3 Competitive International Research Stays.
  - **3 Months** at the University of Texas in Austin, **USA**. Supervised by **Prof. Don Batory (*Mención*)**.
  - **4 Months** at the University of Bristol, **UK**. Supervised by **Prof. Kerstin Eder**.
  - **6 Months** at the KTH, **Sweden**. Supervised by **Prof. Dilian Gurov**
- 8 Research Projects, being 1 International H2020 funded and acting as **Task Leader**:
  - Codenames: HADAS, MAGIC, TASOVA, **DAEMON**, MEDEA, LEIA, RHEA, and IRIS.

Highlighted Thesis Publication 1:



Title: Uniform Random Sampling Product Configurations of Feature Models That Have Numerical Features

Authors: Daniel-Jesus Munoz, Jeho Oh, Mónica Pinto, Lidia Fuentes, Don Batory

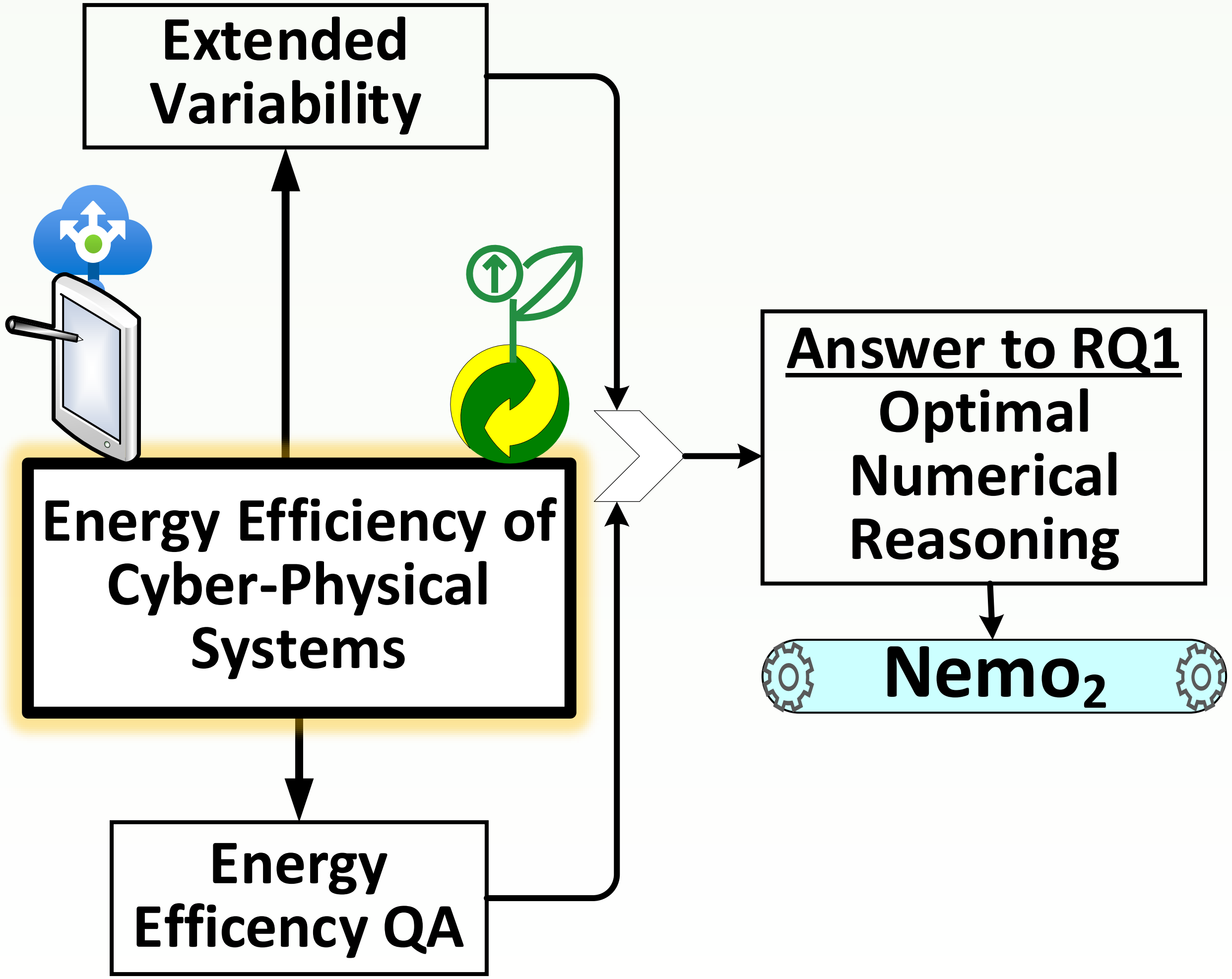
Conference: **23th ACM International Systems and Software Product Line Conference (SPLC 2019)**

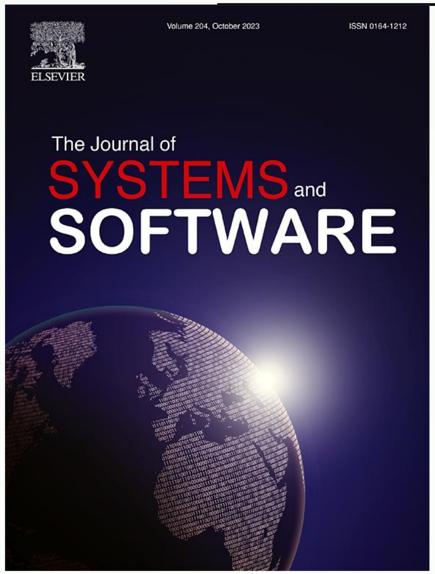
GGS Class: **2**

Publication Date: September 2019

DOI: https://doi.org/10.1145/3336294.3336297

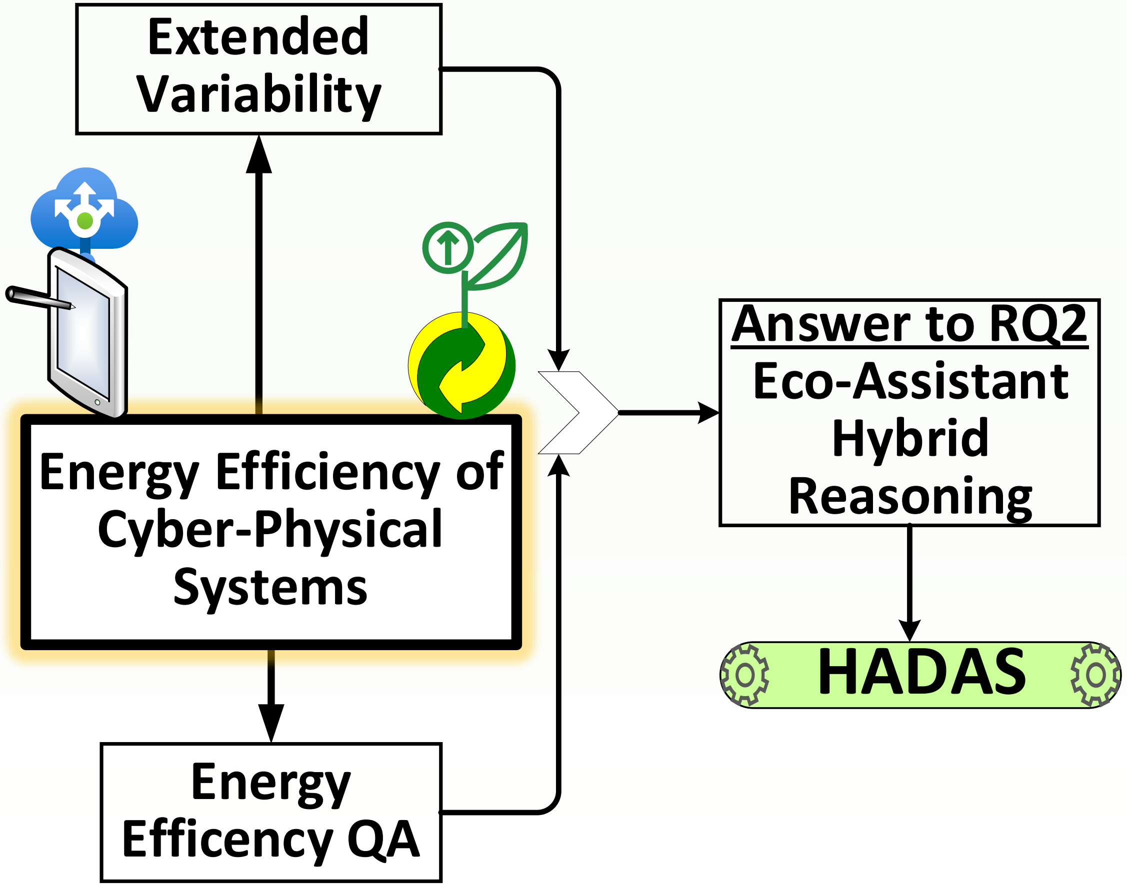
Highlighted Thesis Publication 2:





Title:	Transforming Numerical Feature Models into Propositional Formulas and the Universal Variability Language
Authors:	Daniel-Jesus Munoz, Mónica Pinto, Lidia Fuentes, Don Batory
Journal:	**Journal of Systems and Software (JSS)**
JCR Impact Factor:	**3.514 (Q2)**
Publication Date:	June 2023
DOI:	https://doi.org/10.1016/j.jss.2023.111770

Highlighted Thesis Publication 3:



Title: Finding Correlations of Features Affecting Energy Consumption and Performance of Web Servers Using the HADAS Eco-Assistant

Authors: Daniel-Jesus Munoz, Mónica Pinto, Lidia Fuentes

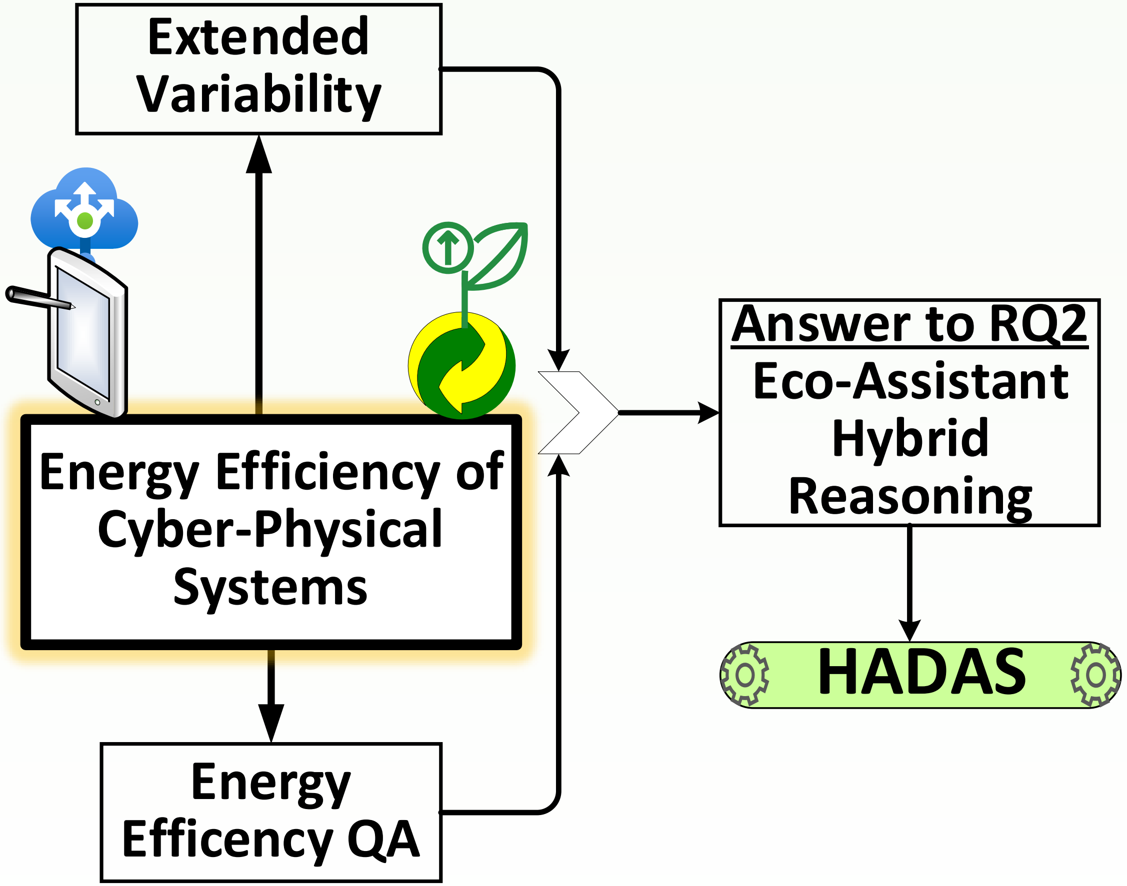
Journal: **Computing**

JCR Impact Factor: **2.063 (Q2)**

Publication Date: June 2018

DOI: https://doi.org/10.1007/s00607-018-0632-7

Highlighted Thesis Publication 4:



Title: Energy-Aware Environments for the Development of Green Applications for Cyber-Physical Systems

Authors: Daniel-Jesus Munoz, José A. Montenegro, Mónica Pinto, Lidia Fuentes

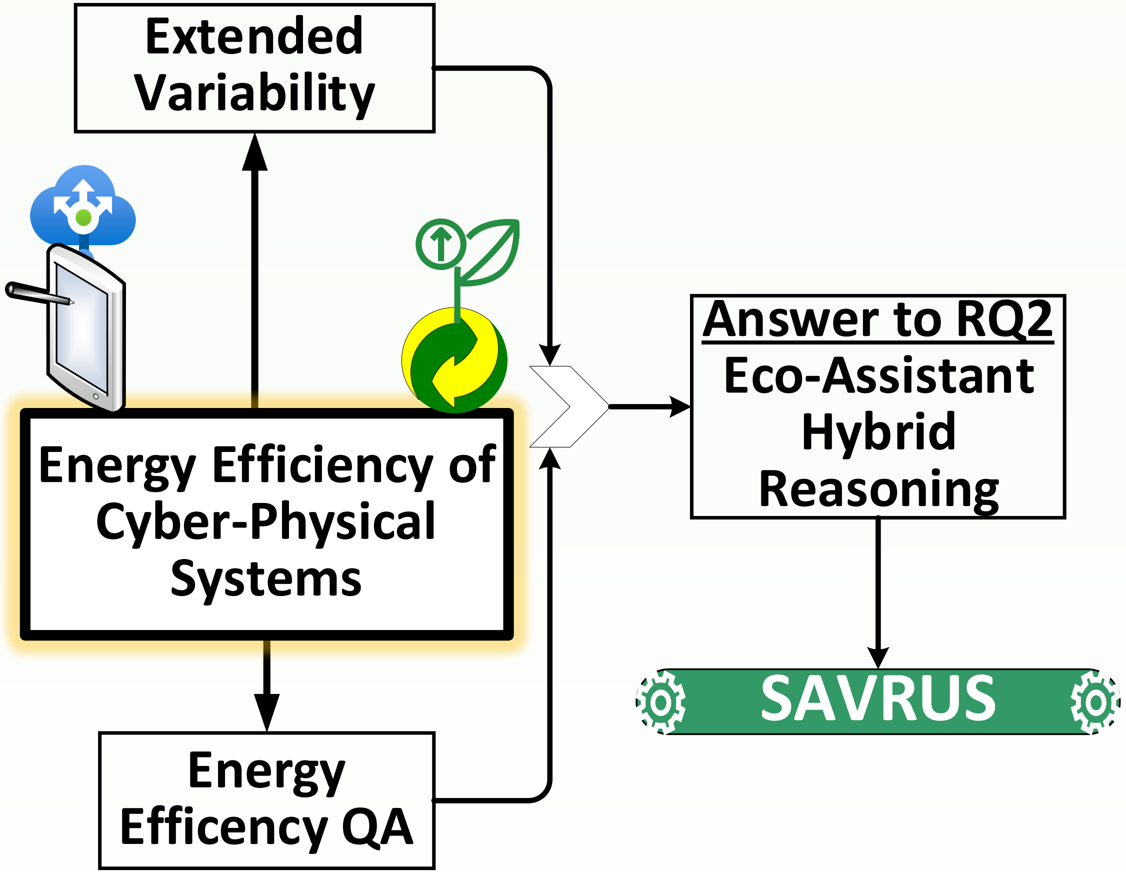
Journal: **Future Generation Computing Systems**

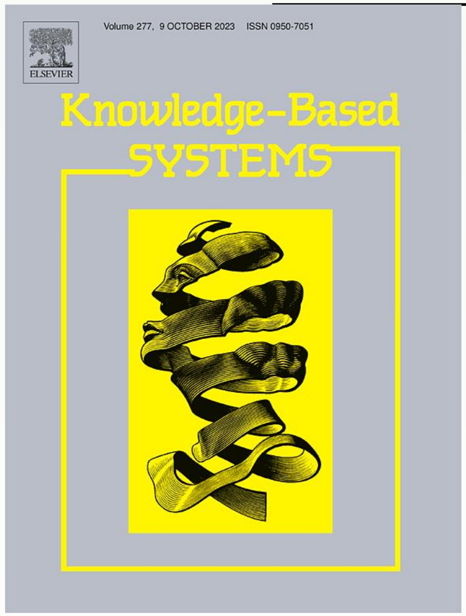
JCR Impact Factor: **6.125 (Q1)**

Publication Date: September 2018

DOI: https://doi.org/10.1016/j.future.2018.09.006

Highlighted Thesis Publication 5:





Title: Detecting Feature Influences to Quality Attributes in Large and Partially Measured Spaces Using Smart Sampling and Dynamic Learning

Authors: Daniel-Jesus Munoz, Mónica Pinto, Lidia Fuentes

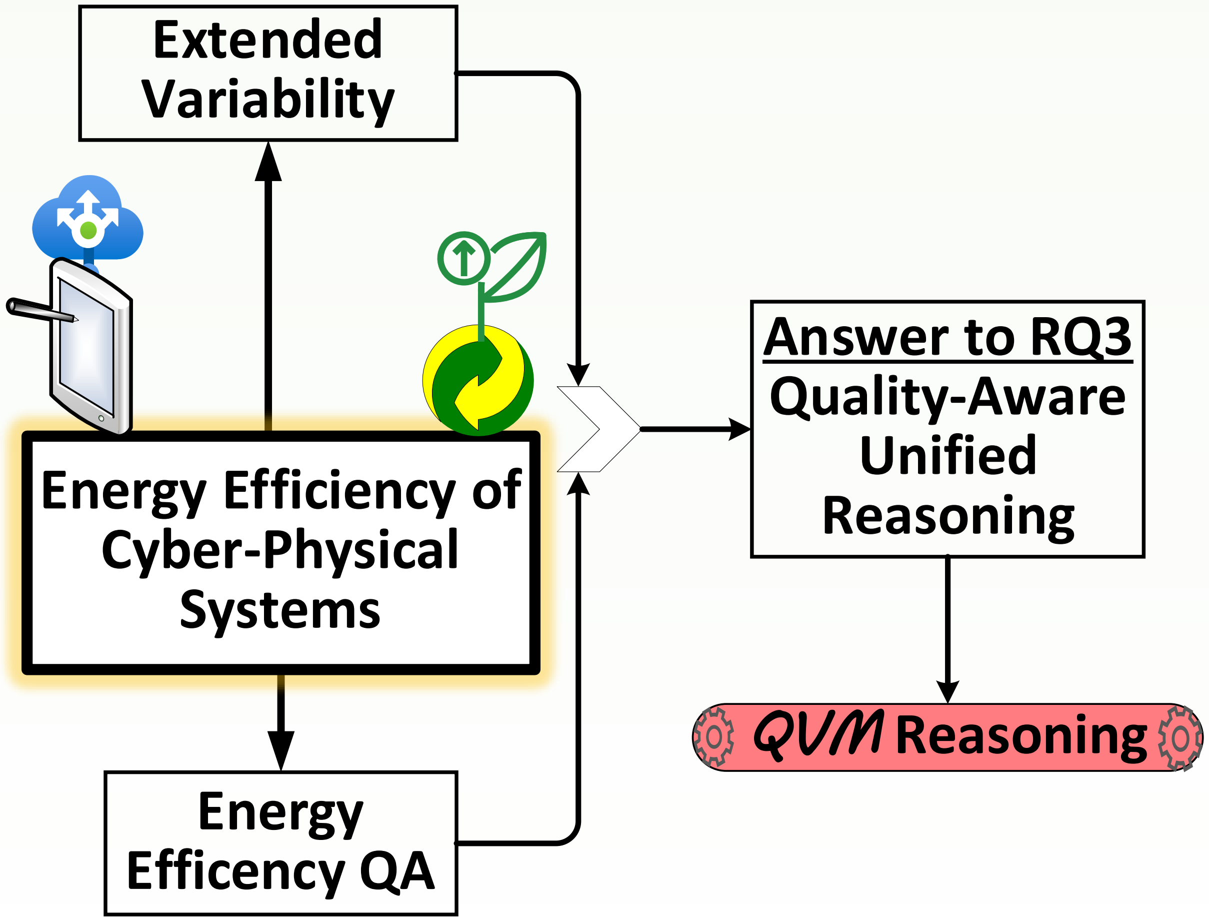
Journal: **Knowledge-Based Systems**

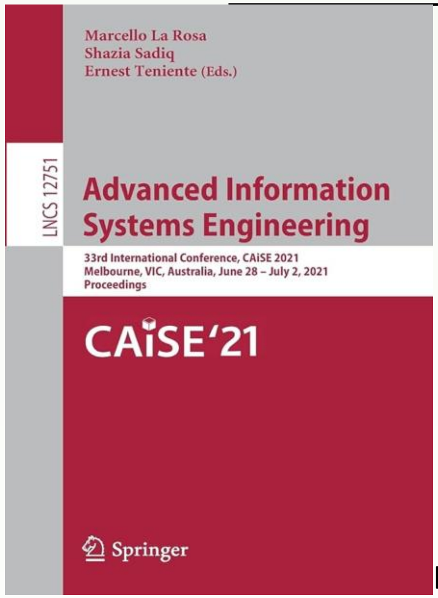
JCR Impact Factor: **8.139 (Q1)**

Publication Date: April 2023

DOI: https://doi.org/10.1016/j.knosys.2023.110558

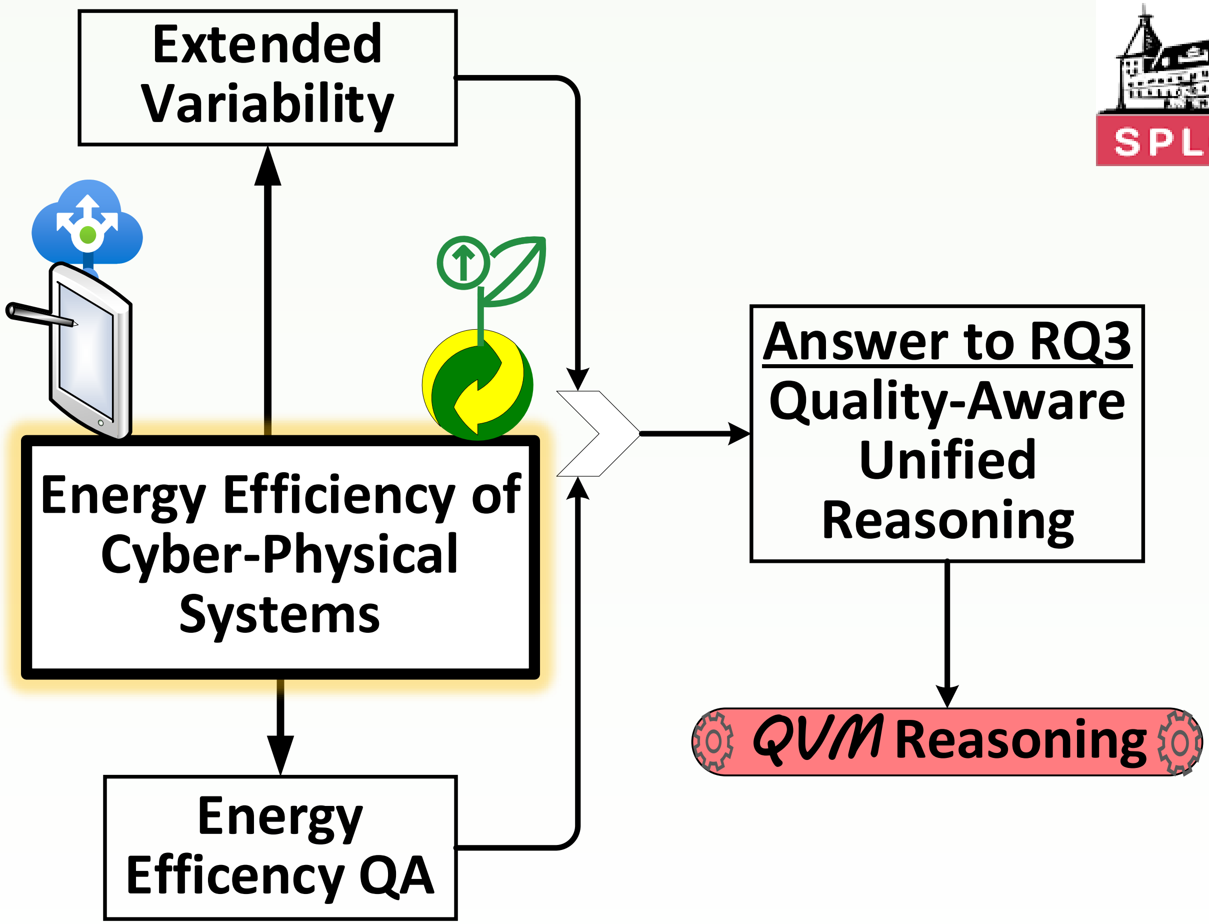
Highlighted Thesis Publication 6:





Title:	Category Theory Framework for Variability Models with Non-functional Requirements
Authors:	Daniel-Jesus Munoz, Dilian Gurov, Mónica Pinto, Lidia Fuentes
nference:	**33rd International Conference on Advanced Information Systems Engineering (CAiSE 2021)**
GGS Class:	**2**
Publication Date:	June 2021
DOI:	https://doi.org/10.1007/978-3-030-79382-1_24

Highlighted Thesis Publication 7:



Title: Quality-aware Analysis and Optimisation of Virtual Network Functions

Authors: Daniel-Jesus Munoz, Mónica Pinto, Lidia Fuentes

Conference: **26th ACM International Systems and Software Product Line Conference (SPLC 2022)**

GGS Class: **2**

Publication Date: September 2022

DOI: https://doi.org/10.1145/3546932.3547007

Conclusions:

1. We can **encode NFs and arithmetic constraints** into bitvectors and PFs to perform efficient optimal configuration search with a minimum performance degradation (e.g., seconds).
2. We can connect the variability with **databases of energy consumption** information with index-based processes. This creates the measured space.
3. If the measured space is colossal, incomplete, and biased, we can follow a **modular sequence** with competitive runtime, coverage and accuracy: solver based sampling, lazy learning, statistical tests, transfer learning and weighted sorting.
4. Category theory can unify the extended NFM and QM as a category, as well as define quality-aware operations for categorical solvers like CQL IDE.

Future Work:

- Integrate machine and transfer learning techniques into our $\mathcal{QVM}$ framework.
- Improve some $SAVRUS$ modules:
  - Learning by self-trained Neural Networks.
  - *k*NN by Approximate Nearest Neighbourgs.
- Replace the relational database by alternatives like non-relational.
- Test other algebras techniques taken from Geometry and Topology.
- Test other algebraic tools like Coq and Agda.
- Formalise calculus for Quality-Aware reasoning like Newton to count.
- Represent and solve variability and quality as systems of equations.
- Migrate our tools and techniques to other areas like distributed systems.

Acknowledgments:

- Supervisors Lidia Fuentes and Mónica Pinto.
- Research stay supervisors Don Batory, Kerstin Eder and Dilian Gurov.
- PhD Thesis Court.
- European Project H2020 DAEMON 101017109
- Spanish Projects :
  - HADAS TIN2015-64841-R
  - MAGIC P12-TIC1814
  - TASOVA MCIU-AEI TIN2017-90644-REDT
  - MEDEA RTI2018-099213-B-I00
  - RHEA P18-FR-1081
  - IRIS PID2021-122812OB-I00
  - PRE2019-087496 (FPI contract)
  - LEIA UMA18-FEDERJA-15

All good?

- Daniel-Jesus Munoz
  dm@luma.es

- **CAOSD** Group (http://caosd.lcc.uma.es/)
  Universidad de Málaga
  ETS Ingeniería Informática.
  Laboratory 3.3.3.

UNIVERSIDAD
DE MÁLAGA

CAOSD