Extended Variability Models, Algebra and Arithmetic

Daniel-Jesus Munoz, Mónica Pinto and Lidia Fuentes
ITIS Software, CAOSD , Universidad de Málaga,
Andalucía Tech, Spain

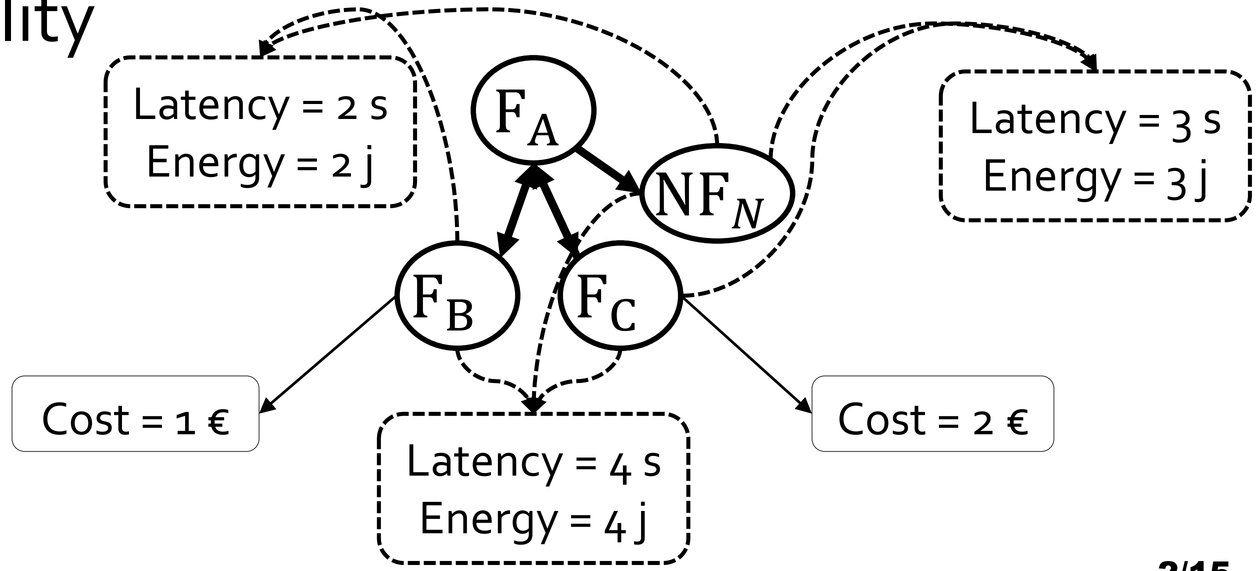
FOSD'23
Ulm, Germany. March 2023

Extending Feature Modeling and Reasoning

1. Numerical features and arithmetic constraints
2. Feature-wise Quality Attributes (QAs) for simple aggregation functions
3. Variant-wise QAs
4. Complex constraints involving any type of feature and QA
5. Customisable reasoning: < Optimise: 0.7 Cost, 0.3 Energy Rate (i.e., j ÷ s) >
6. Enumerated and multiple cardinality
7. Nested and hierarchical Qas
8. Multi-feature (clonables)
9. Etc.

Aggregation: Addition ($Cost_i$)

$2*Cost < Latency + Energy$



Modelling and Reasoning Support (1/2)

Table 1
Language constructs supported by feature modeling tools (ter Beek et al., 2019; Horcas et al., 2022, 2020).

Tools	Optional feat.	Xor group	Or group	Abstract feat.	Mutex-Group	Group Cardinality	Multi decomp.	Multi-feature	Typed feature	Numerical feat.	Feat. attribute	Binding time	Default value	Delta value	Range	Simple const.	Prop. log. const.	First-order const.	Relational expr.	Arithmetic expr.	Type const.	Default const.	Compositions	Conf. reference	Containers	Model version	Multi-views	Configuration	Partial conf.
Glencoe	●	●	●	○	○	●	○	○	○	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	◐	○	●	●
SPLOT	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	●	◐	○	○	○	○	○	○	○	○	○	○	●	●
FaMa	●	●	●	○	○	●	○	○	○	○	●	◐	○	○	○	●	◐	○	○	○	○	○	○	○	○	○	○	●	●
Clafer	●	●	●	●	●	●	○	●	●	●	◐	◐	◐	○	◐	●	●	●	●	●	◐	●	◐	◐	◐	○	○	●	●
FeatureIDE	●	●	●	●	○	○	○	○	○	○	◐	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	●	●
pure::variants	●	●	●	●	○	○	○	◐	○	◐	◐	○	○	○	○	●	●	○	◐	◐	○	○	●	◐	◐	○	○	●	●

Table 2
Current tool support for automated analysis of feature models (Benavides et al., 2010; Horcas et al., 2022).

Tools	Void feat. model	#Products	Dead features	Valid product	All products	Explanations	Refactoring	Optimization	Commonality	Filter	Valid conf.	Atomic sets	False optional	Corrective explan.	Depend. analysis	ECR	Generalization	Core features	Variability factor	Arbitrary edit	Cond. dead feat.	Homogeneity	LCA	Multi-step conf.	Roots features	Specialization	Orthogonality	Redundancies	Variant features	Wrong cardinal.
Glencoe	●	●	●	●	○	●	○	○	●	○	●	●	●	●	●	●	○	●	●	○	●	●	○	●	○	○	●	●	●	●
SPLOT	●	●	●	●	○	○	○	○	●	○	●	○	○	○	○	●	○	●	●	○	○	○	○	●	○	○	●	○	○	●
FaMa	●	◐	●	●	◐	●	○	○	◐	○	◐	○	○	●	●	○	○	◐	◐	○	○	○	○	○	○	○	○	○	●	○
Clafer	●	◐	○	●	◐	○	○	◐	○	○	◐	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
FeatureIDE	●	◐	●	●	◐	●	○	○	○	○	●	●	●	●	●	●	○	●	◐	○	●	○	○	●	○	○	●	●	●	●
pure::variants	●	◐	○	●	○	●	○	○	○	○	●	○	○	●	●	○	○	○	○	○	○	○	○	●	○	○	●	○	○	●

●: support. ○: not support. ◐: support with limitations or not scale for large models.

Modelling and Reasoning Support (2/2)

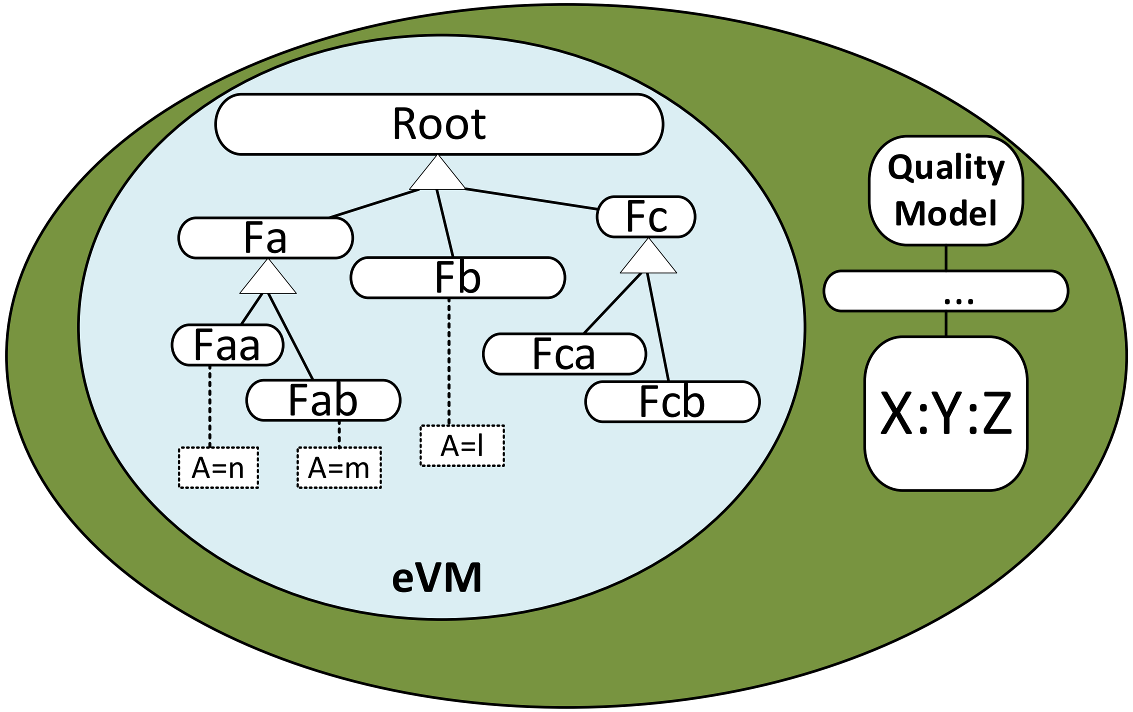
Munoz, D. J., Pinto, M., & Fuentes, L. at SPLC (2022, September).

	T:ClaferMoo	T:FAMA	T:FeatureIDE	T:pure::variants	T:SPL Conqueror	T:QAMTool	T:HADAS	T:STEAM	A:SATIBEA	A:MILPIBEA	A:GIA	A:MO-DAGAME
Quality Modelling												
Feature-level QAs (Quality information is linked to individual features)	■	■	■	■	■	■	□	■	■	■	■	■
Configuration-level QAs (Quality information is linked to valid configurations)	□	□	□	□	▒	■	■	□	□	□	□	□
Formalising and Solving Variability Models with Qualities												
Declarative Paradigm (CSP solver, BDD solver, SAT solver, ...)	■	■	■	■	□	□	□	□	■	□	■	□
Declarative Paradigm + Additional Assets (SAT solver + database, ...)	□	□	□	□	■	■	■	□	□	□	□	□
Search-Based Software Engineering + FM representation	■	□	□	□	■	□	□	□	■	■	■	■
Alternative Formalisation (Abduction/Deduction reasoning, Category Theory)	□	□	□	□	□	□	□	■	□	□	□	□
Automatic Quality Reasoning												
Model Analyses Operations (satisfiability, count features, count configurations, ...)	■	■	■	■	■	■	■	■	░	░	░	░
Aggregation Function Operations												
* Addition	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
* Product	Q	□	□	Q	Q	□	Q	□	□	□	Q	□
* Mean	□	□	□	Q	□	□	□	□	□	□	□	□
* Approximation arithmetic equation	□	□	□	□	Q	□	□	□	□	□	□	□
Optimal Search Operations												
* Maximum	Q	Q	□	□	Q	□	□	Q	Q	Q	Q	Q
* Minimum	Q	Q	□	□	□	□	□	Q	Q	Q	Q	Q
* Multiobjective	Q	□	□	□	□	□	□	□	Q	Q	Q	Q
* Range optimisation	Q	□	□	□	Q	□	□	□	□	□	□	□

A Category Theory Vision of Different Algebras

- *Focus on what is common while abstracting from the specifics*
- A category is form by objects, Objects are related by Arrows, and categories are related by Functors.
- Some examples:
  - Category of Sets (i.e., Set Theory)
  - Category of Relations (i.e. Relational Algebra)
  - Category of Topos (i.e. Deformed Geometry)
  - *Category of Software Product Lines*

Model: Entity Relationship for Relational Databases



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**

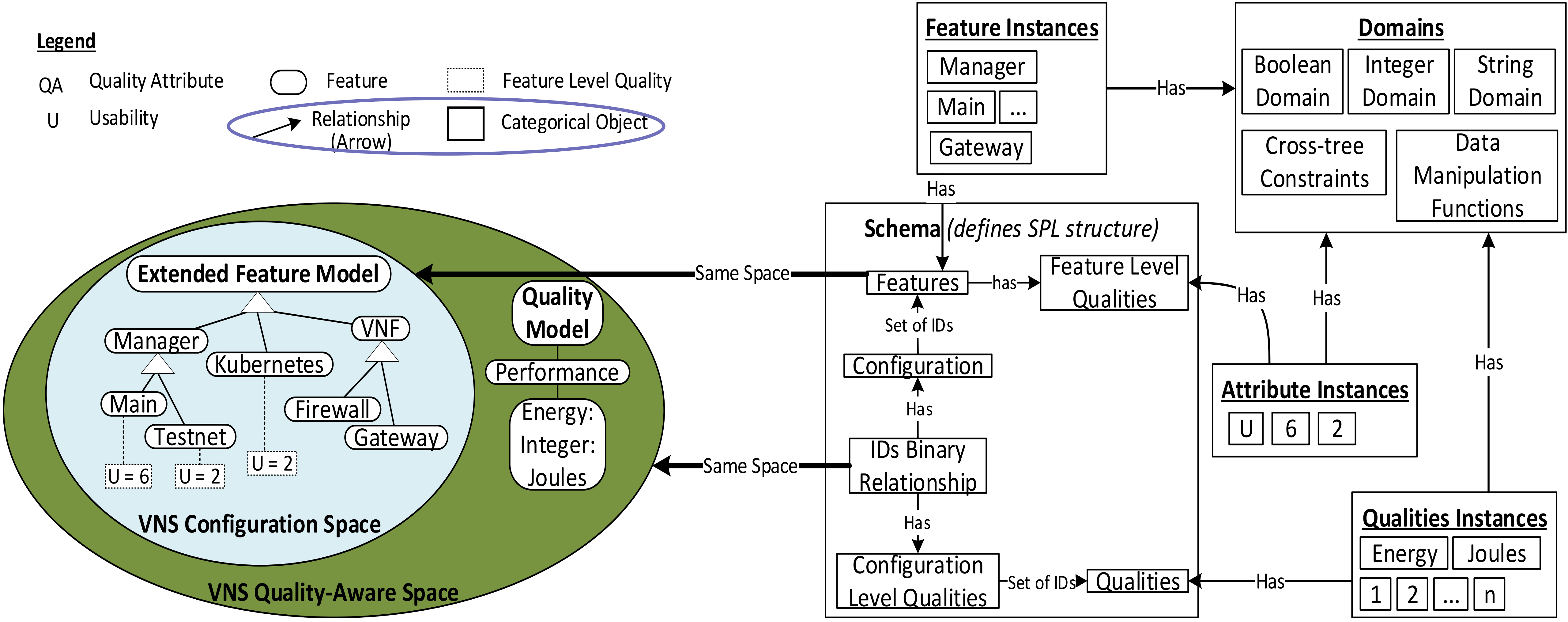
ID	Name	Domain	Parent
-	X	Z	-*
…			

Feature Attribute

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

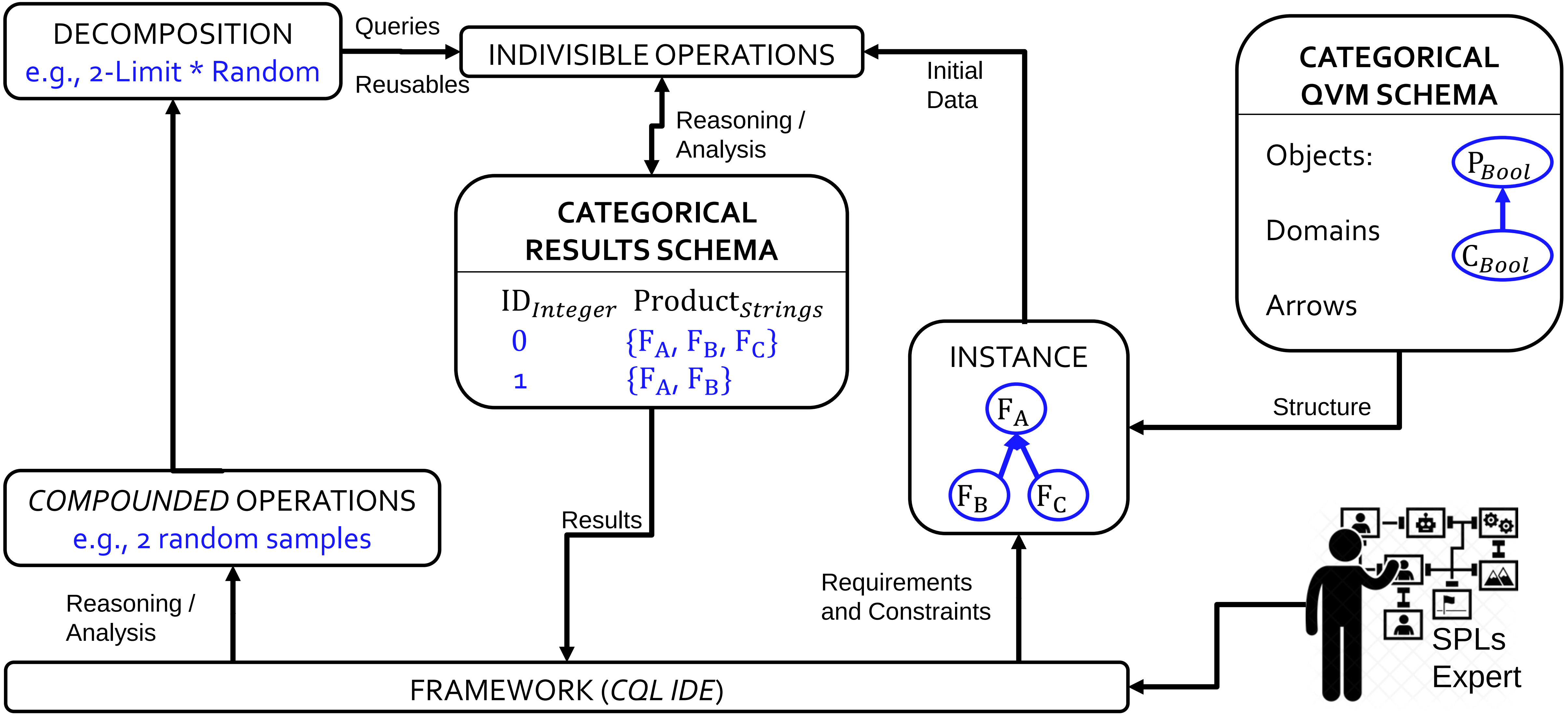
Daniel-Jesus Munoz, Monica Pinto, and Lidia Fuentes @ SPLC'19
"Hadas: analysing quality attributes of software configurations"

Model: The SPL Category

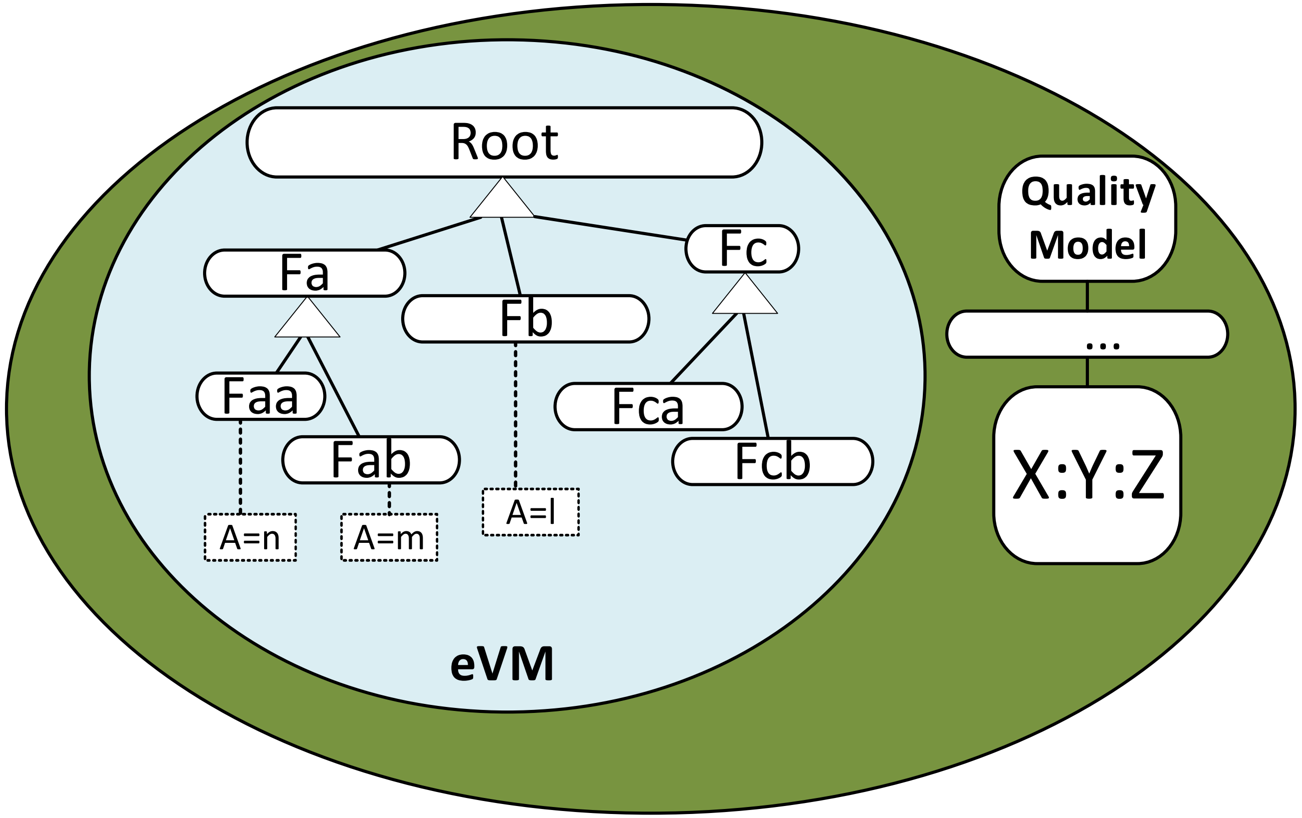


Daniel-Jesus Munoz, Monica Pinto, and Lidia Fuentes @ SPLC'22
"Quality-aware analysis and optimisation of virtual network functions"

Reasoning: A CQL IDE Framework for the SPL Category



Model: System of Linear Inequalities (CSP-ILP)



$-1 <$ Root|Fa|Faa.value|Fab.value|Fb.value|Fc|Fca|Fcb < 2

Fa + Fb + Fc = Root

Faa + Fab = Fa

Fca + Fcb = Fc

Faa.A = m

Fab.A = n

Fb.A = l

"In process…"

Feature models, Grammars, and Propositional Formulas (Batory 2005)

(Classic) Feature Model ≡ Propositional Formulas

Now we apply a theoretical computer science perspective:

1. Propositional Logic ≡ A model of Boolean Algebra
2. Boolean Algebra ⊆ First-order Logic Algebra
3. First-order Logic Algebra ≡ Boolean Category
4. Boolean Category ⊆ Heyting Category [...]

(Classic) Feature Model ≡ A model of Heyting Category

Intuitionistic Feature Models?

Its restrictions produce proofs that have the existence property

Available Proof Assistants: Agda or Coq

Objectives

1. To increase the modeling and reasoning tools at our disposal
   - E.g., Database Systems, CQL IDE, the complete Matlab suite, etc.
2. To apply other theories reasoning over extended VMs
   - E.g., Geometry automated reasoning, Numerical analysis, etc.
3. To improve extended VMs reasoning
   - E.g., Horner scheme (SAT), Newton (Approximate Counting), ILP, etc.
4. To uncover new (useful) properties of extended VMs
   - (Partial) Derivativation, integration, limits, determinant...?

Lessons Learned From my Experience

- Feature Modelling tools:
  - Are easier to maintain
  - Has a good user support and a united community
  - Tend to *deprecate*
- Multi-purpose methodology and tools:
  - Are harder to maintain
  - Have the potential to do more but needs expertise to perform
  - Community is of a diverse background
  - Have a (steep) initial learning curve

Other Efforts in the Literature

- Reasoning of propositional formulas represented as equations
  - F. Lin; From Satisfiability to Linear Algebra; 2013
- Theory-generic algorithm for variant-based satisfiability
  - J. Meseguer; Variant-based satisfiability in initial algebras; 2017
- Categorical transformations of software product lines.
  - G. Taentzer et al.; Transformations of software product lines: A generalizing framework based on category theory; 2017
- Solving systems of equations over finite algebras
  - P.M.Idziak; Satisfiability in Multi-valued Circuits; 2018

Inspiration: Alexander Grothendieck

- Berlin 1928. Half Jewish => captured for concentration *camps*
- After the 2nd World War he studied Mathematics, and specialised in Topological Spaces and Functional Analysis
- Soon applied to Quantum Physics and Computer Science:
  - Grothendieck Inequality ≈ Einstein-Podolsky-Rosen Paradox
- Goal: To unify disparate mathematical fields together
- In 1972 published the K-Theory and the Grothendieck universe
- Disappeared in 1992, and burned/hide his new theories
- Hidden in a small village in the Pyrenees till death (2014).

Dankeschön!