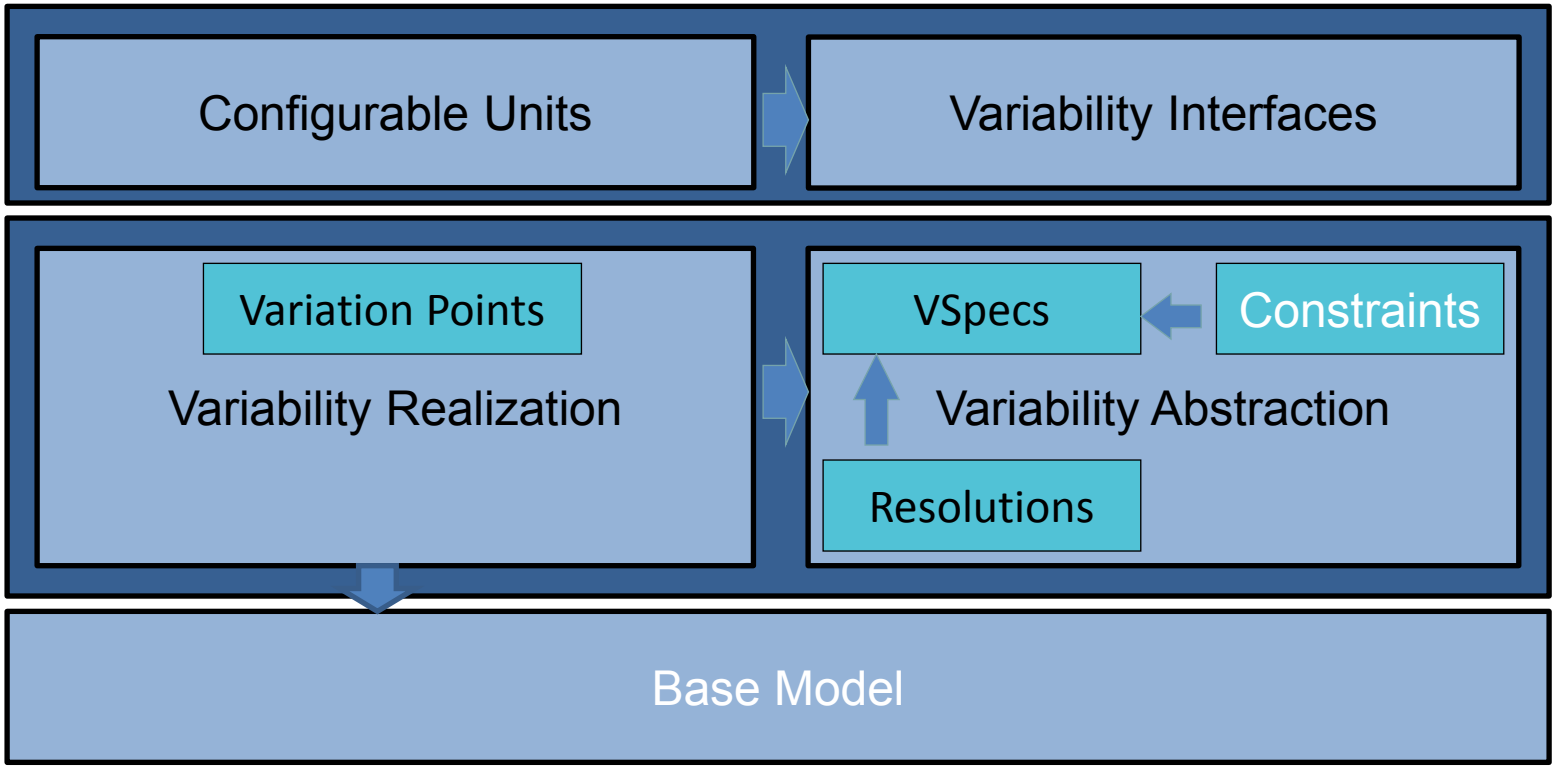
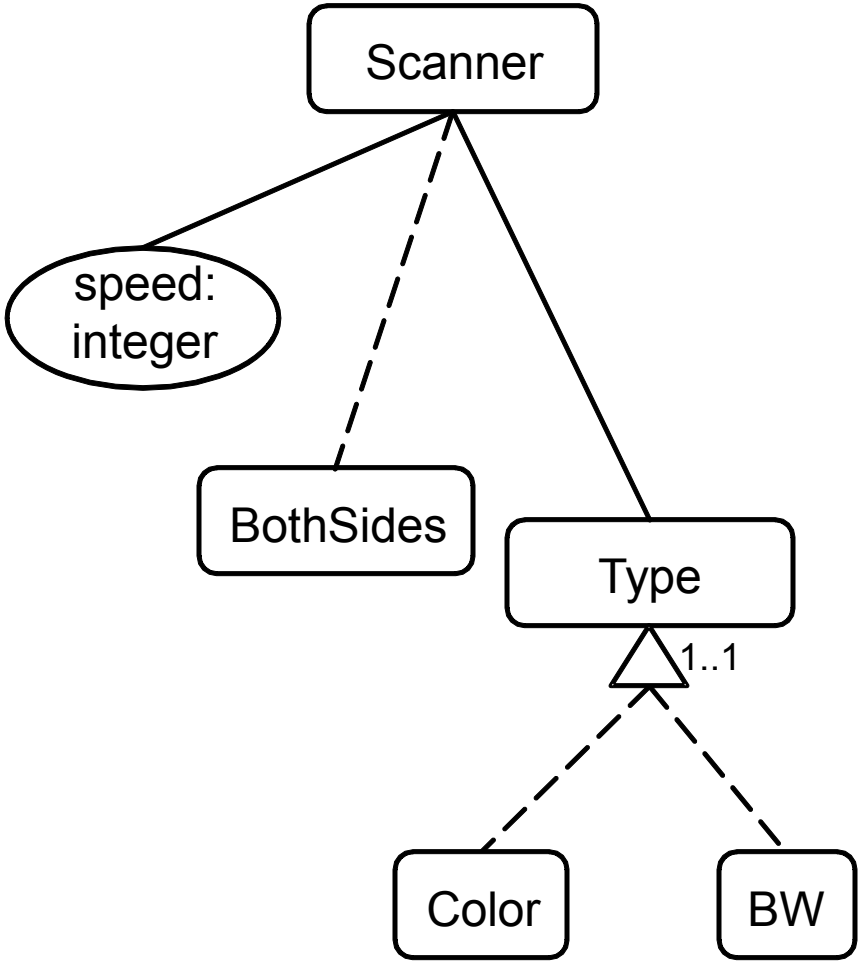
CVL Tutorial

Details of Abstraction

Recall CVL Architecture

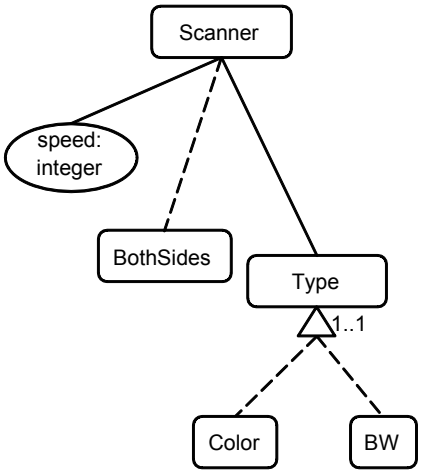


Variability Specification (Vspec)



Variability Specification (Vspec)

- Abstract variability specifiers
- Similar to features in feature modeling
- Essentially **decisions** like in decision modeling
- Example: “GPS” is a feature of a camera, but it is also a choice, which can be decided yes/no
- VSpecs can be used for feature modeling
- Variation points are bound to VSpecs, giving semantics to VSpecs.
- VSpecs are arranged in trees,
- Parent-child relations organize the resolution space in usual way, like in FMs.

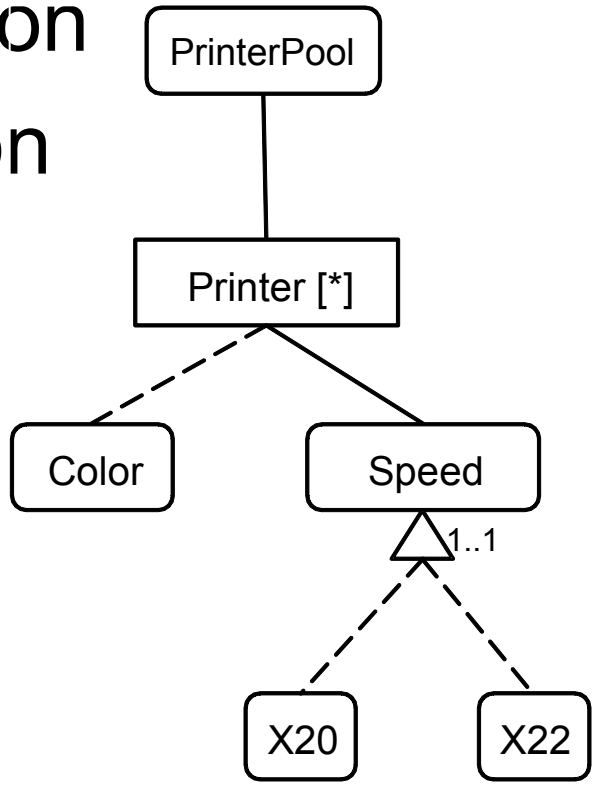


VSpec Types (continued)

- **Choice** requires yes/no decision
- **Variable** is a VSpec allows for providing a value of a specified type
- **VClassifier** allows for creating instances and then providing per-instance resolutions for the VSpecs in its sub-tree
- **Composite VSpecs** – used for modularity; explained later

Tree Structure Semantics

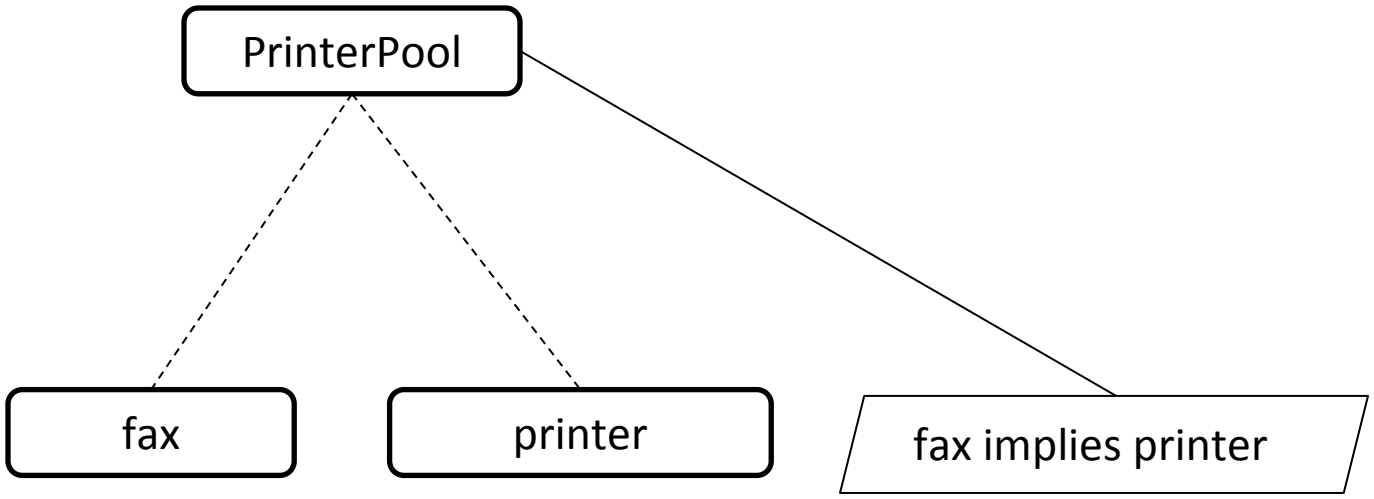
- Configuration semantics akin to cardinality based feature models
- Negative resolution implication
- Positive resolution implication
- Group Multiplicity
- Instance Multiplicity



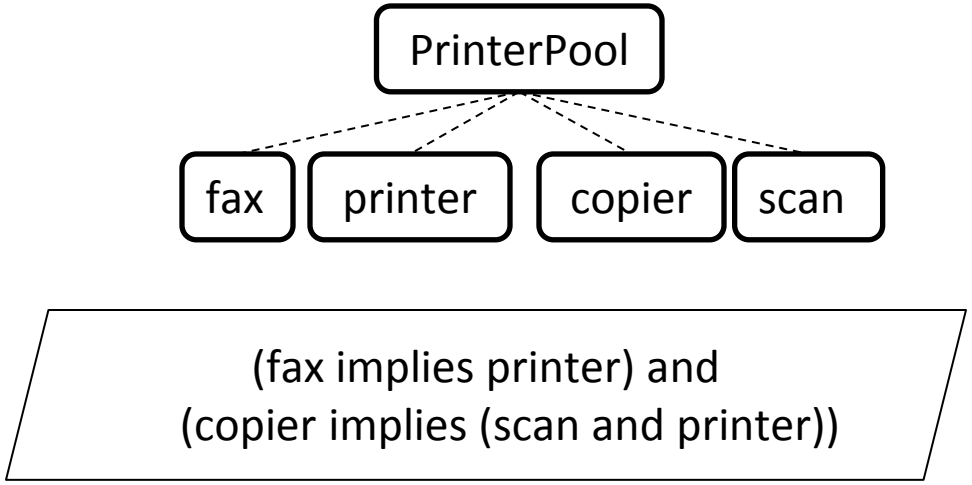
Constraints

- Constraints express intricate relations between VSpecs
- Relations that cannot be captured by trees
- CVL offers basic constraint language
- A sublanguage of OCL
- Other constraint languages, including full OCL, are admitted.
- We show the basic language by example
- And a bit of the full OCL

Simple Propositional Constraints

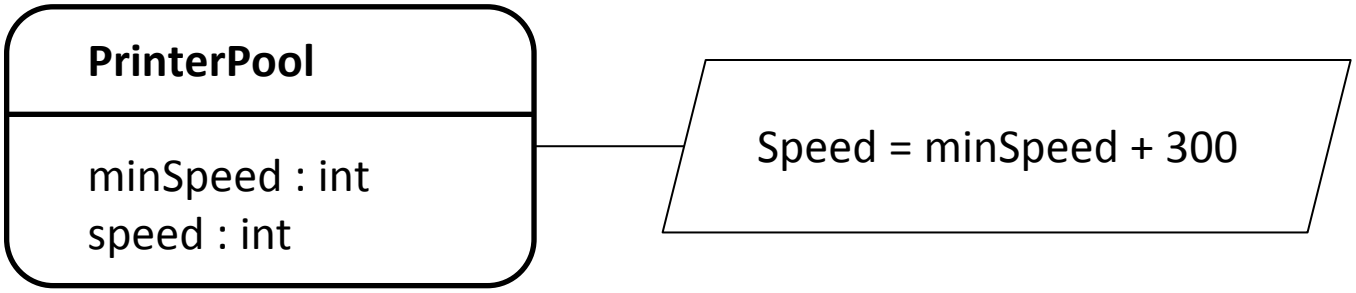


Complex Propositional Constraints

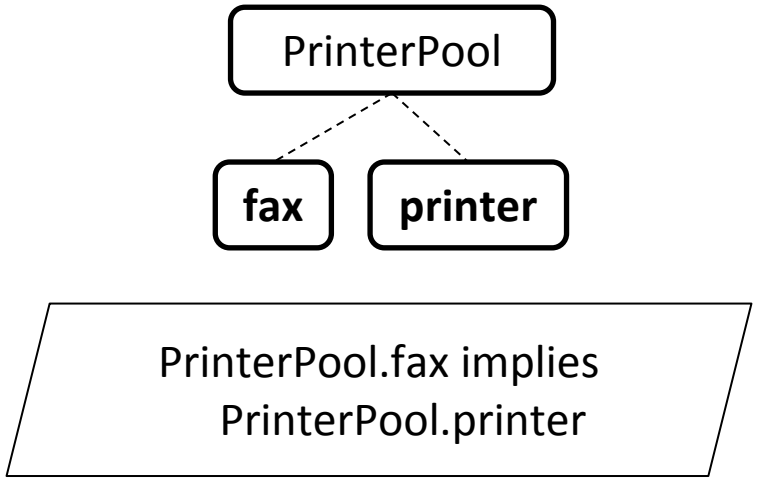


- Global context used in this example
- Using PrinterPool would be equivalent

Arithmetic Constraints

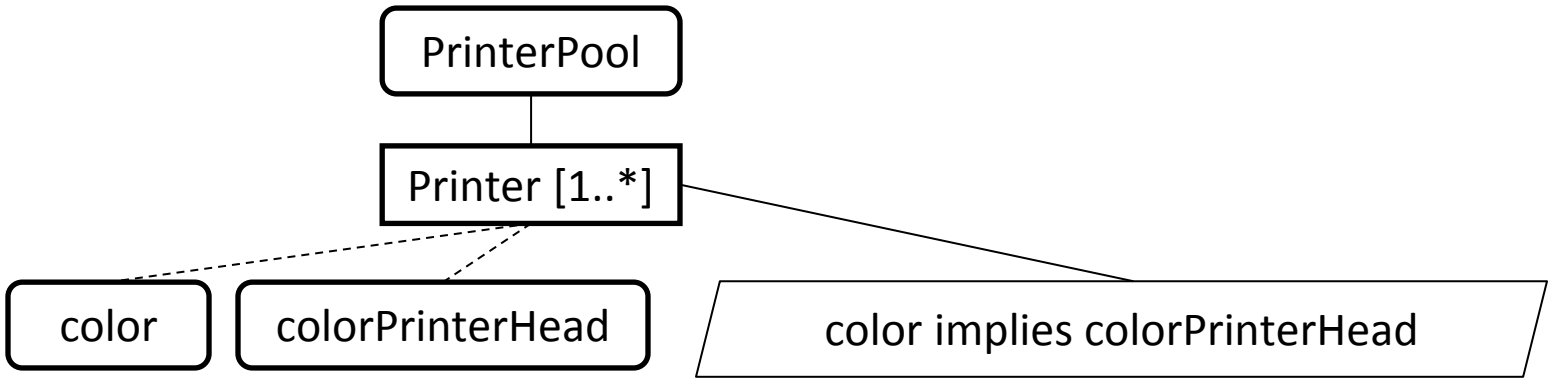


Path Expressions

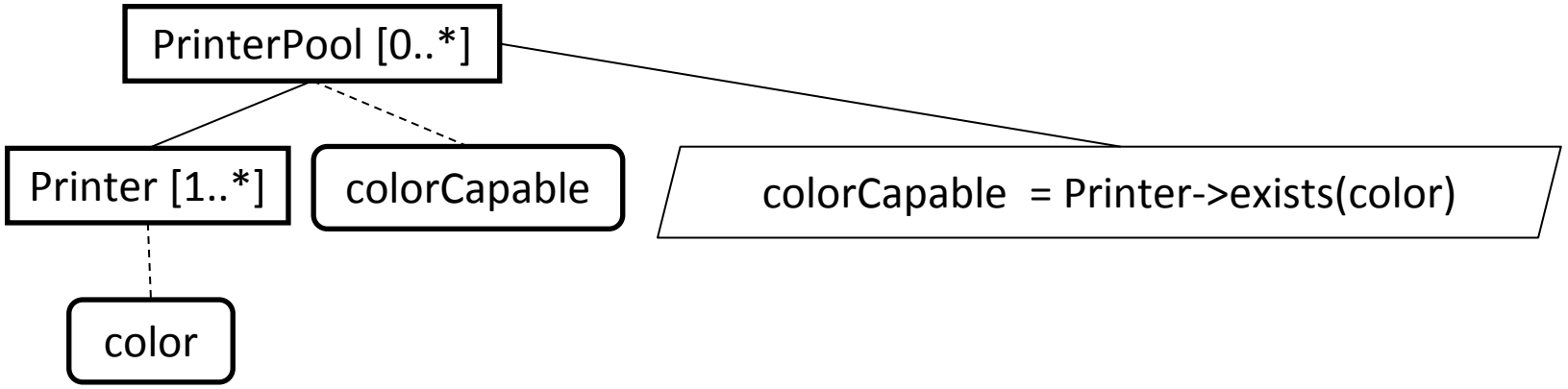


Now using fully qualified names, instead of relying on name disambiguation.

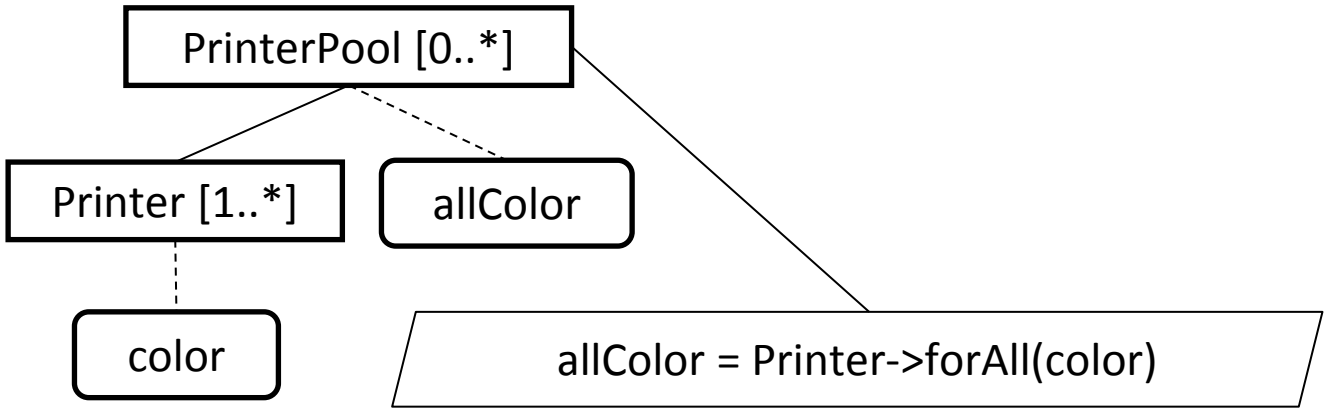
Implicit Quantification



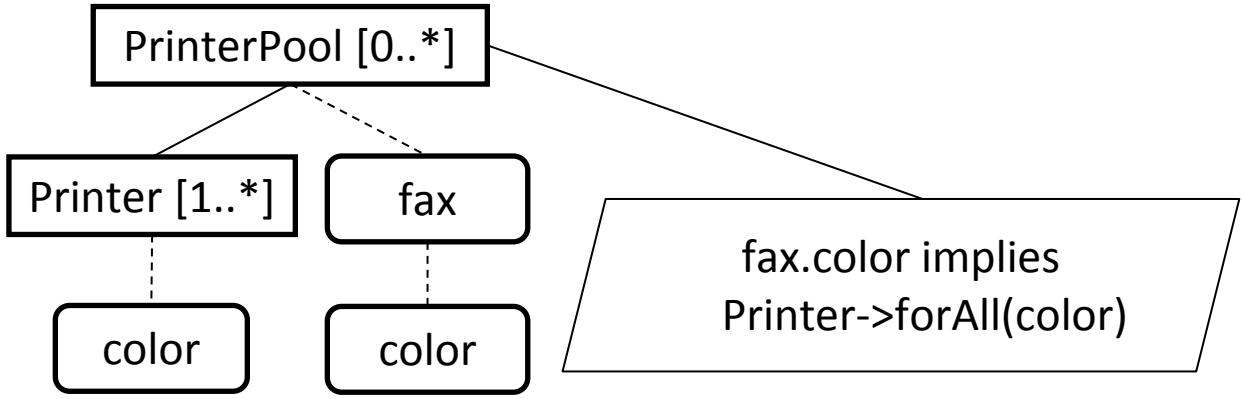
Existential Quantification (Full OCL)

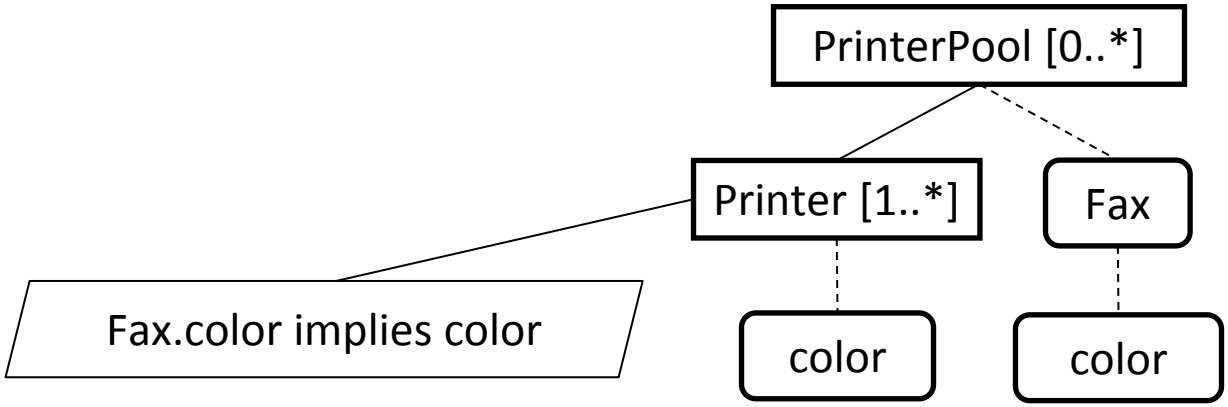


Univ. Quantification (Complete OCL)

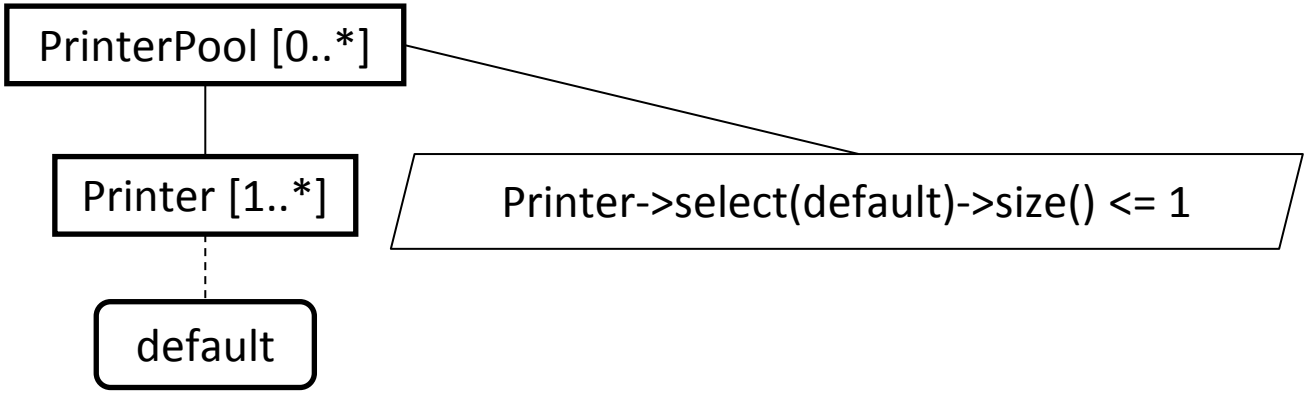


Use Context to Simplify Quantifiers

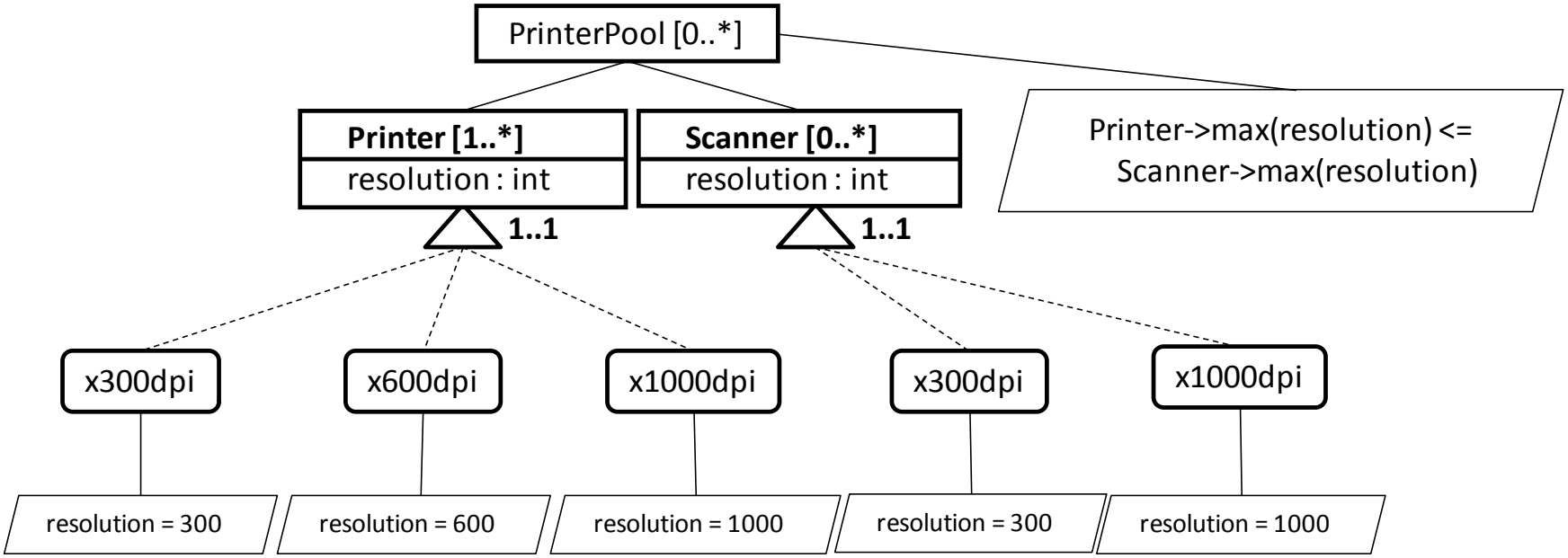




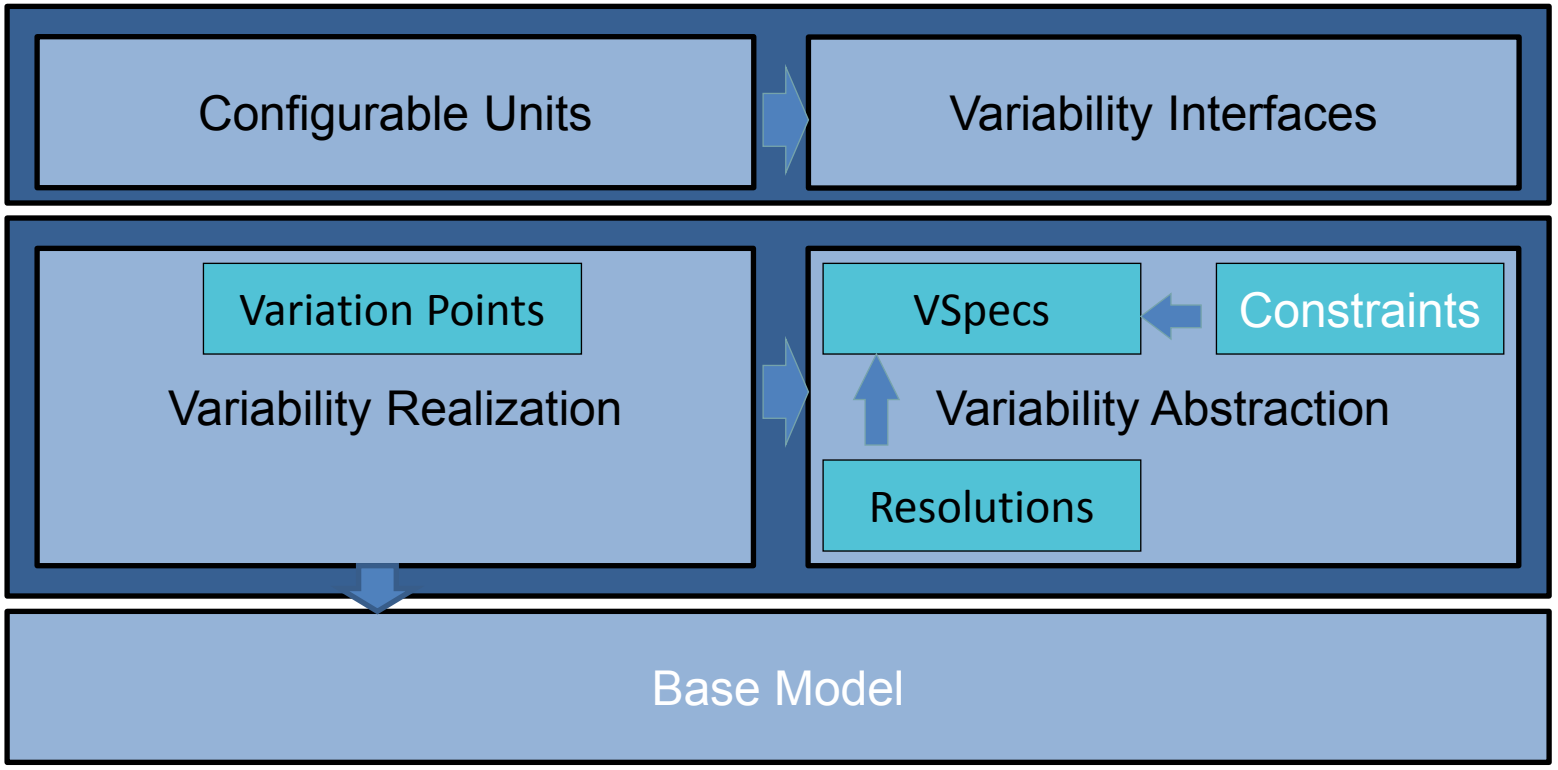
Cardinality Constraints



Arithmetic Constraint w/ VClassifier



Recall CVL Architecture



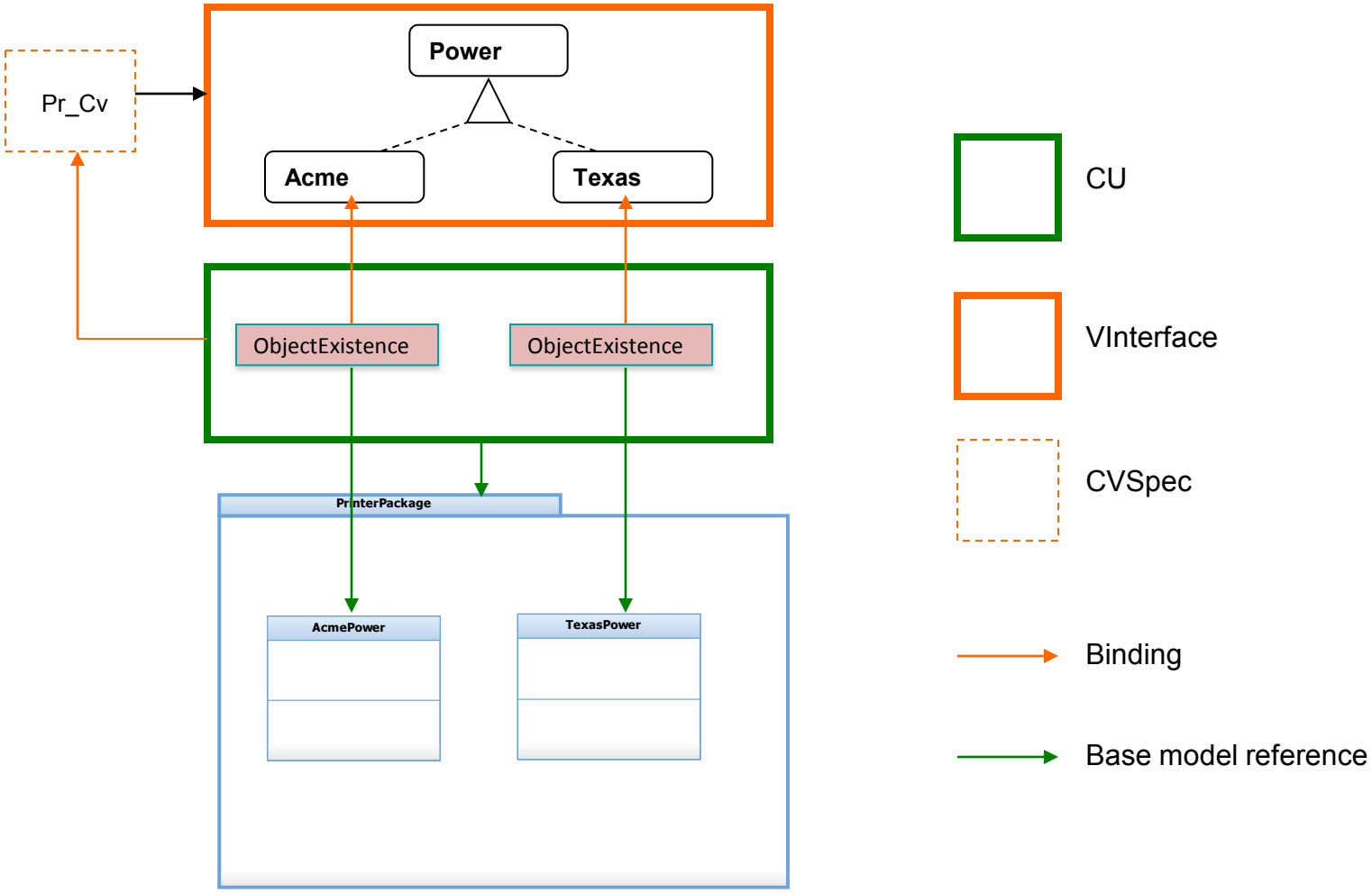
Configurable Units: Mission Statement

- Produce configurable components reusable across multiple projects
- Associate a collection of variability declarations with a base model container
- Containers: UML/SYSML component, package or class, etc …
- Hide details, Expose variability interface
- Component can be configured
- Reuse CUs: clone-and-own or by reference

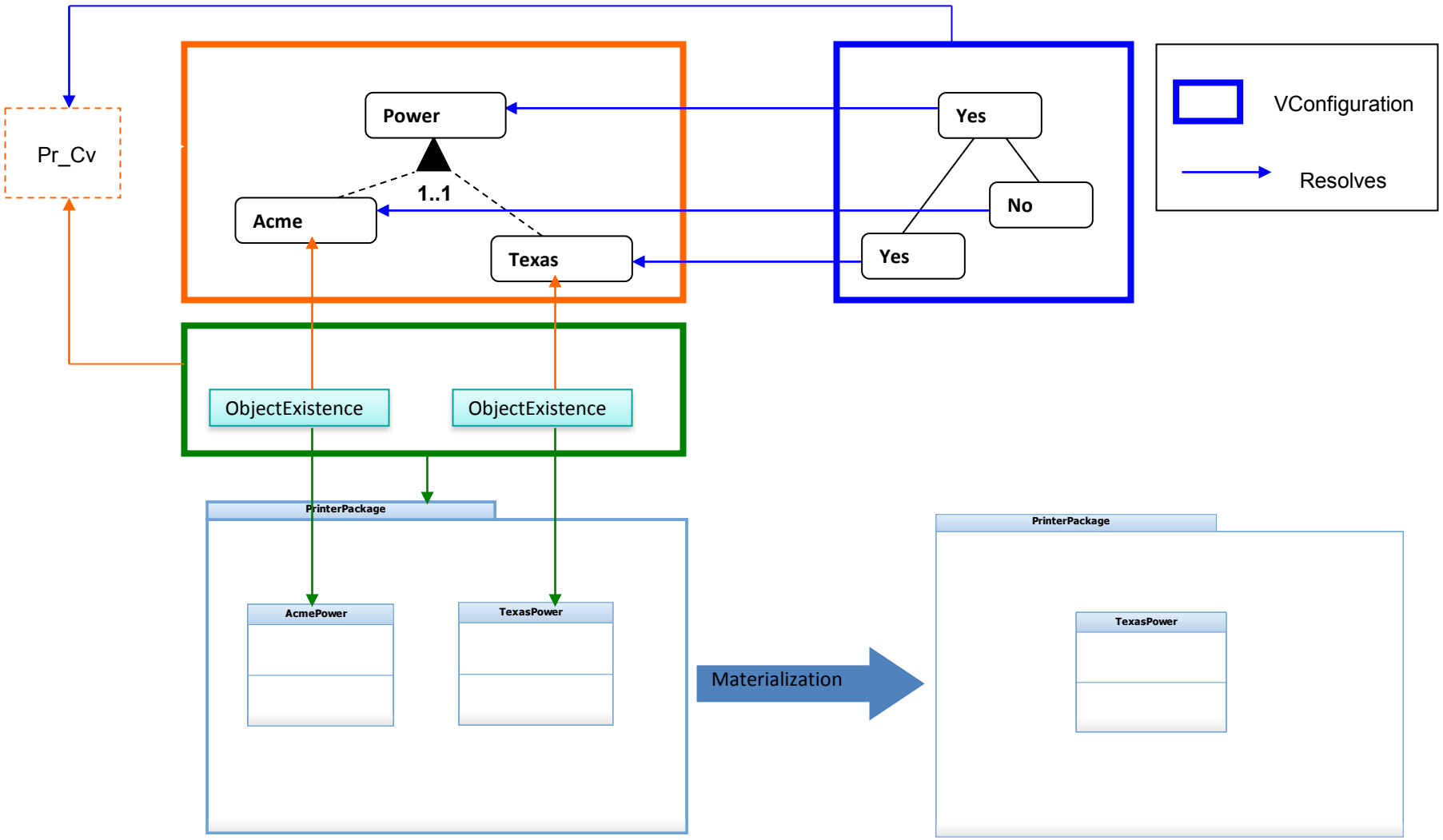
Configurable Unit: How?

- Technically a new kind of variation point
- Entire CU is a single variation point
- References an entire object of base model, which becomes a variability container
- CU contains other variation points
  - inner variability
  - *Composite Variation Point*

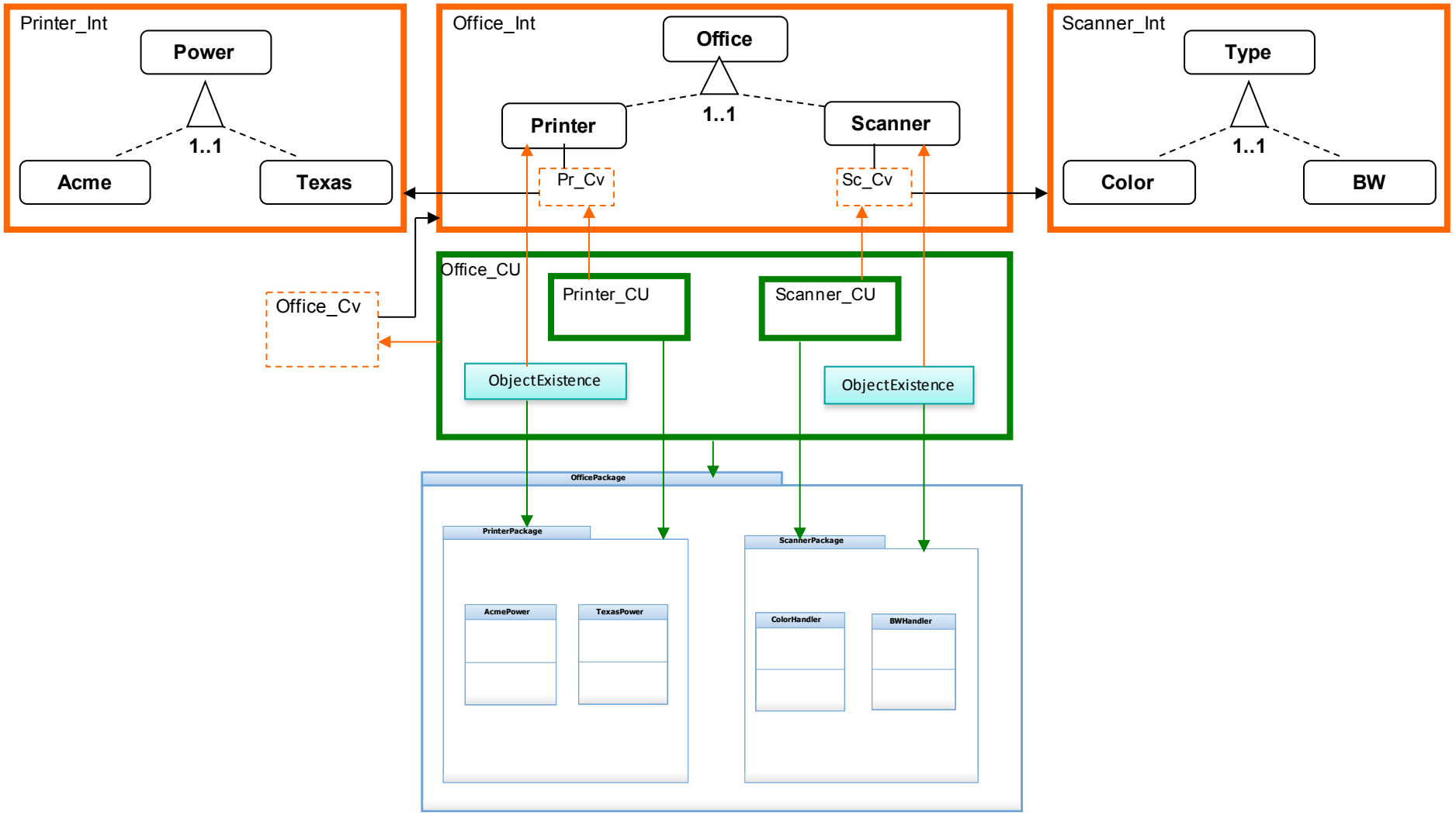
Configurable Unit and VInterface



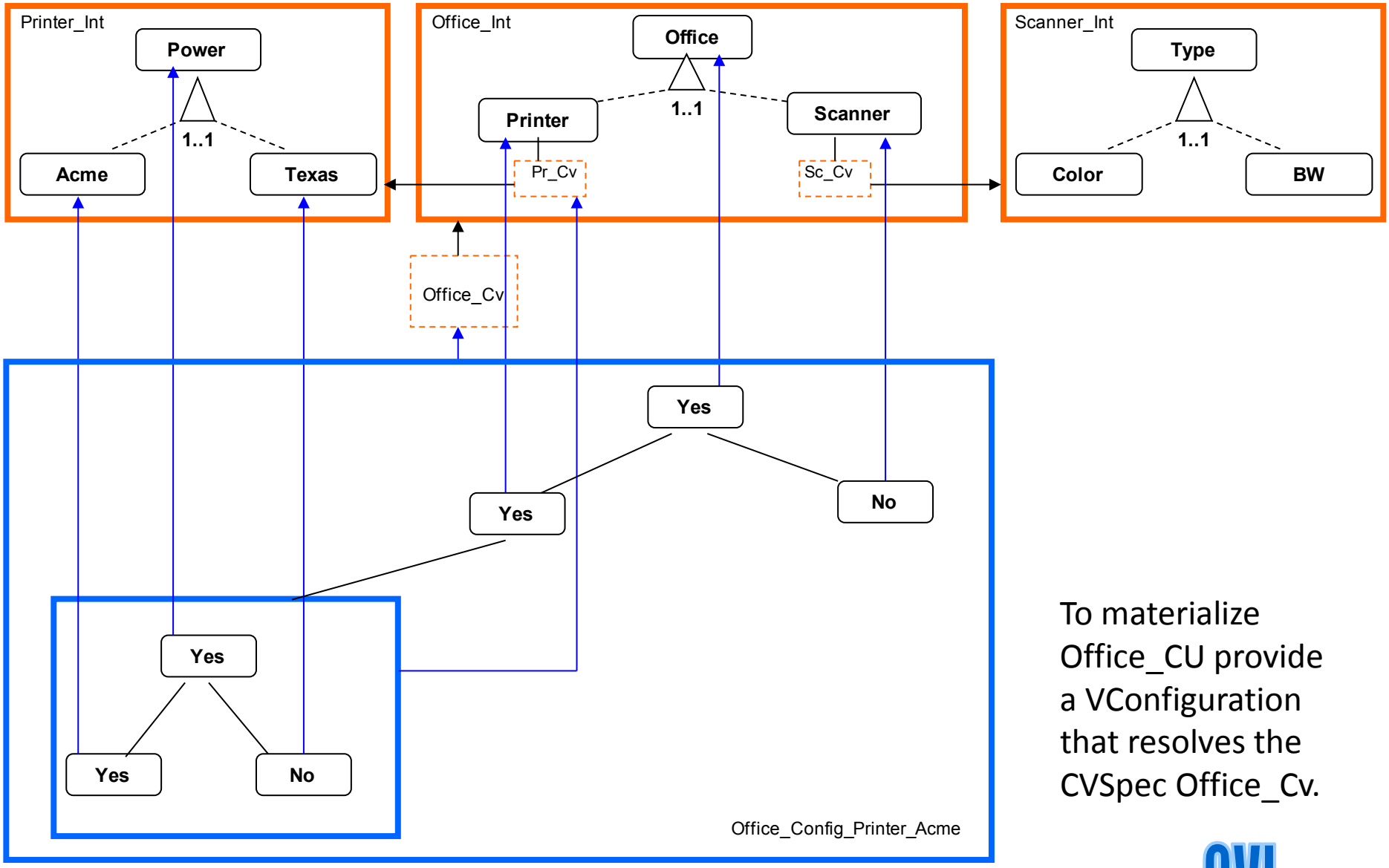
VConfiguration



Composition of Configurable Units

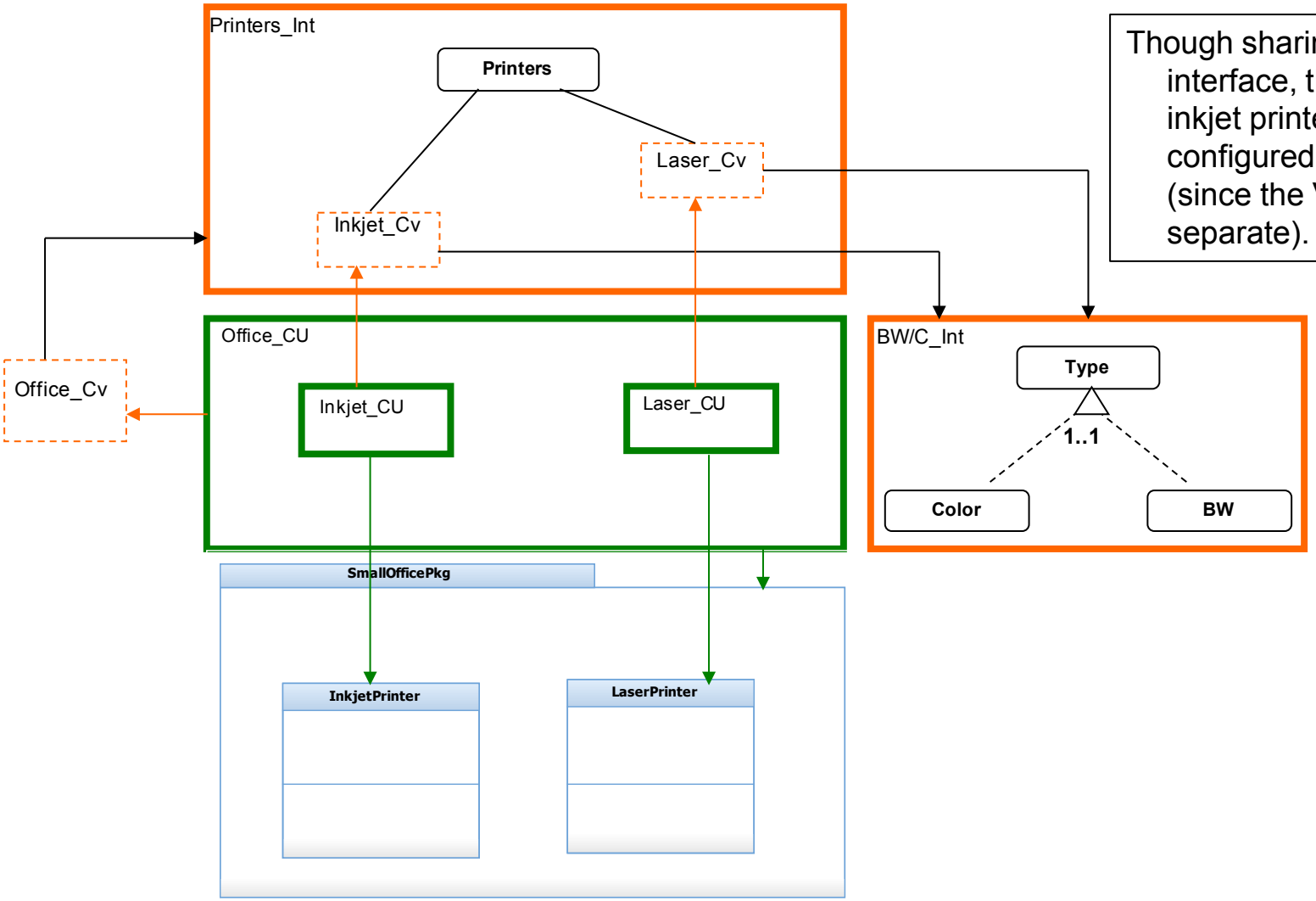


Resolution of Complex CU



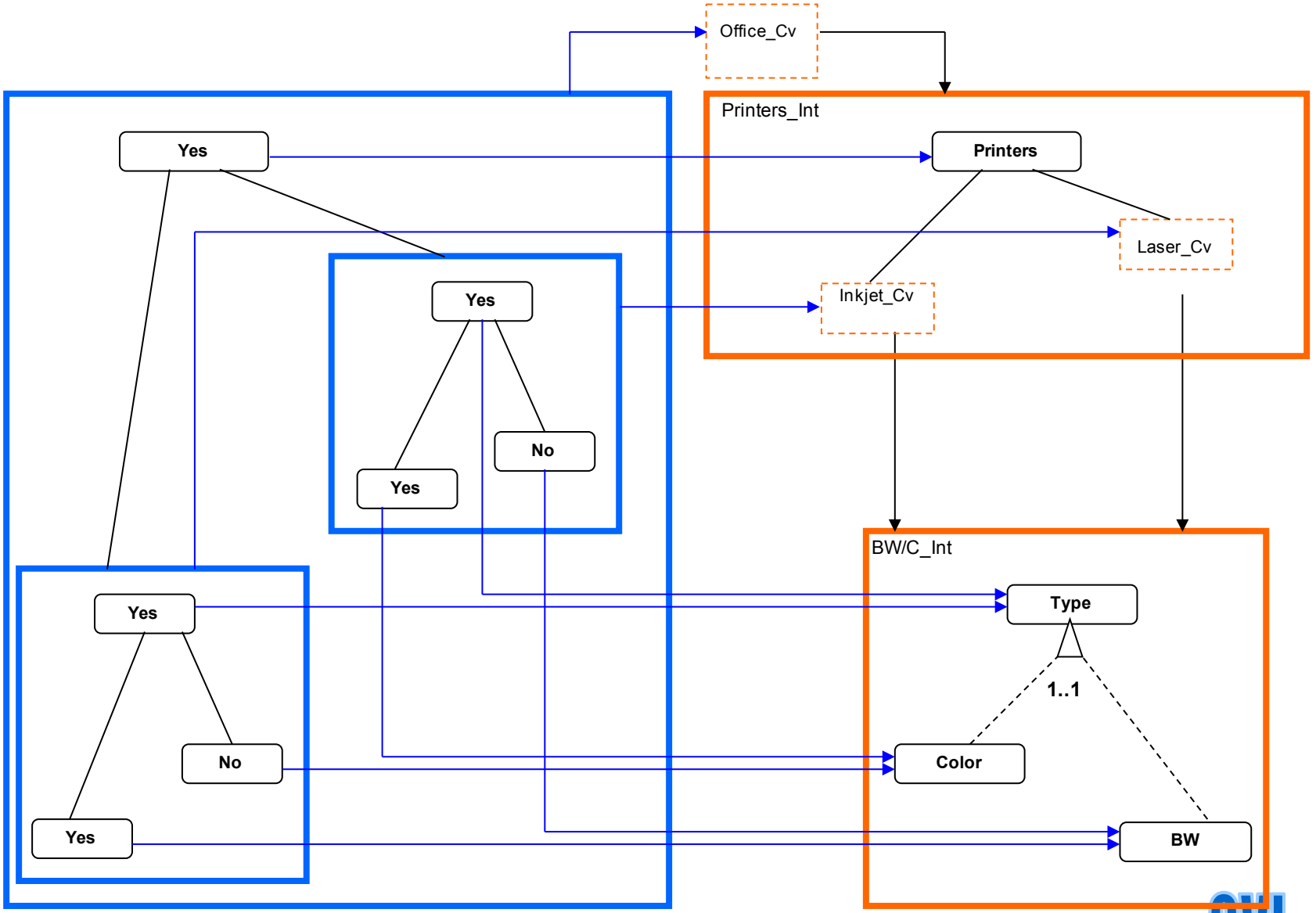
To materialize Office_CU provide a VConfiguration that resolves the CVSpec Office_Cv.

Reusing Variability Interfaces



Though sharing the same interface, the laser and inkjet printers are configured separately (since the VSpecs are separate).

Resolution in Reuse Scenario



Briefly on CVL Experiments

SINTEF CVL Tool

INRIA Implementation

- The semantics in the submission is supported by a Kermeta implementation
- Relying on EMF
- Developed independently of SINTEF

ITU Experiment

- 4 students with a few years of professional experience
- Used about 328 hours to implement about half of the specification
- With no prior experience of MDD
  - steep learning curve
- Estimated 2-3 PMs for the core language
- More if constraints from scratch
- This is without any concrete syntax editor

CLAFER Wiki Uses CVL Syntax

- Interactive modeling environment (WIKI)
- Combines natural language reqs w/ models
- For Example-Driven Modeling (EDM)
- Using Clafer
- Does analyses of Clafer Models
  - Feature models
  - Class-like models
- Visualizes Clafer feature models using CVL concrete syntax

CLAFER Wiki Uses CVL Syntax