

VARiability In safety-critical Embdeded Systems



ARTEMIS-2011-1
Project Number: 295397

CVL –

Common Variability Language or Chaos, Vanity and Limitations?

by Øystein Haugen, SINTEF

Overview of the talk



- Short bio of Øystein Haugen
- What is CVL?
- Chaos
- Vanity
- Limitations
- The Future of CVL



University and Research Inst.

- 80-81: UiO, Research assistant for Kristen Nygaard
- 81-84: Norwegian Computing Center
 - **Simula**-machine
- 97: Practitioners' verification of SDL systems (Dr. scient.)
- 98-03: Ifi, UiO as Part time Associate Professor
- 04-07 : Associate Professor at Ifi (100%)
- 07- : Senior Researcher SINTEF
 - Projects on modeling languages e.g. for variability, train control and pay rolls
- 07- : Assoc. Professor at Ifi (20%)
- 10: General Chair MODELS 2010

Industry and Standardization

- 84-88: SimTech, typographical applications
- 88-90: ABB Technology
 - **SDL**, prototype SDL tool, ATC
- 91-97: Independent Consultant
- 96-00: Rapporteur ITU for **MSC**
- 97-03: Ericsson, NorARC
- 99-11: OMG wrt. **UML 2.0**
 - Responsible for UML 2.x chapter on Interactions
- 09 - : OMG **CVL** – Common Variability Language
 - Coordinates joint submission team



What do we mean by “variability”?



- Product Line variance
 - often variants of the same software base
- Cross-cutting variability
 - often variability is orthogonal to the software design
 - variability needs are discovered after the first software design
- The variability designer is not always the software designer
 - division of labor and of competences



Common ways to model variability



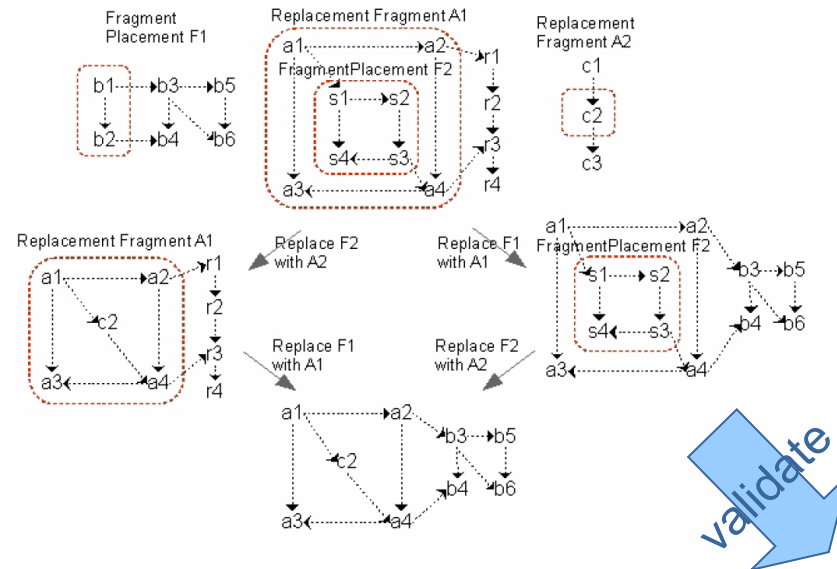
	Framework/ Configuration	Union-of-all- systems	Domain Specific Languages
How?	By mechanisms of a general language	As annotations to a language	By the specific language mechanisms
Constructs	Function, Type, Inheritance, Template, Plugin	Pragma, Stereotype	Proprietary language constructs
unforeseen modeling needs	Just enhance the final model	Enhance the product line model	If not expressible, enhance the language



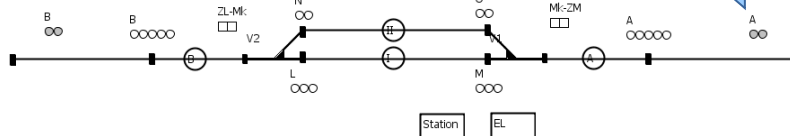
CVL - Common Variability Language



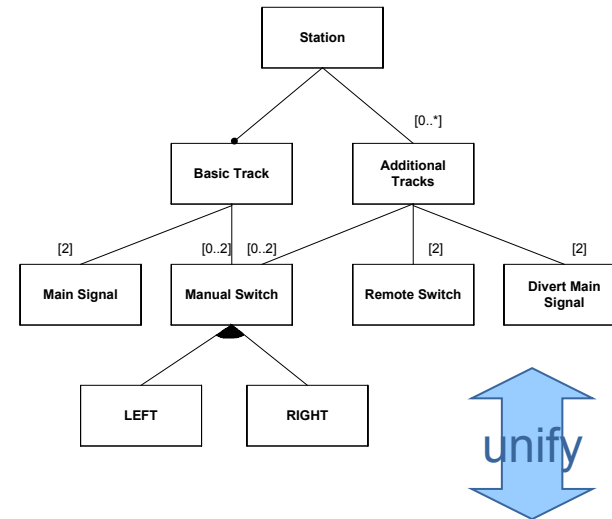
Formal Analysis



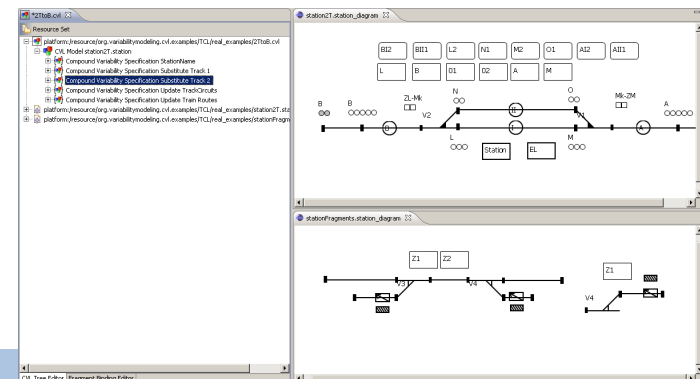
Base: DSL e.g. TCL



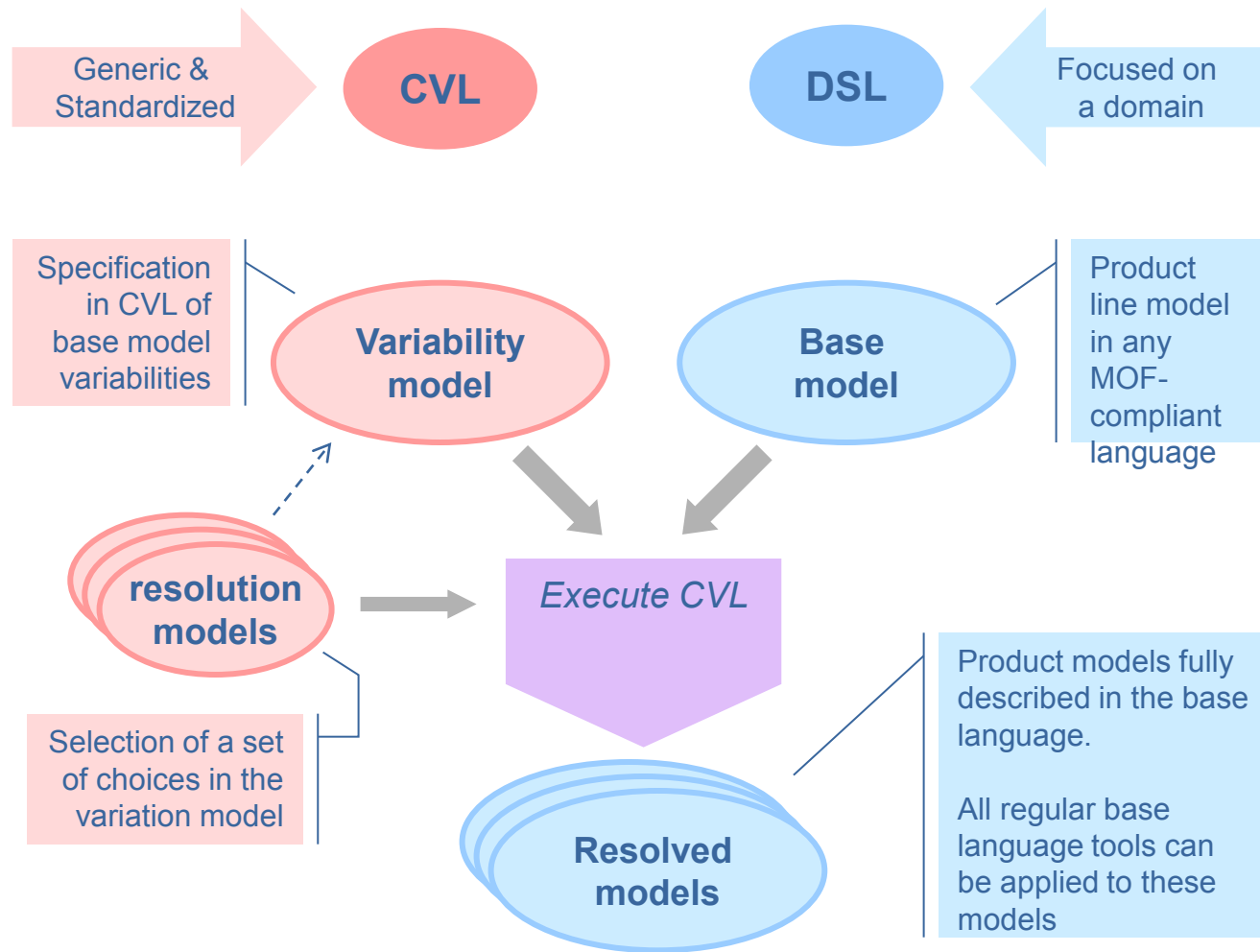
CVL Model

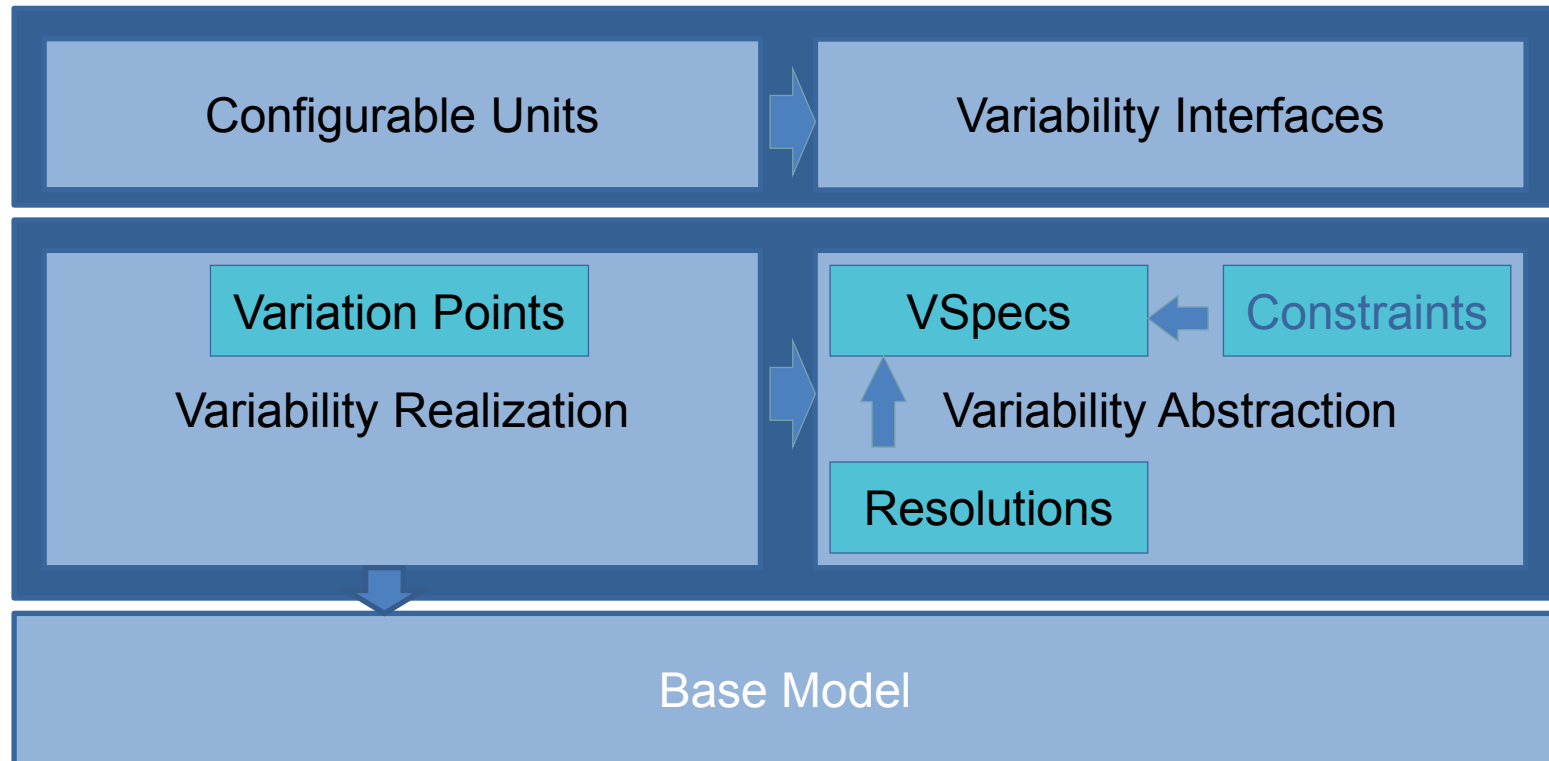


CVL and its tool

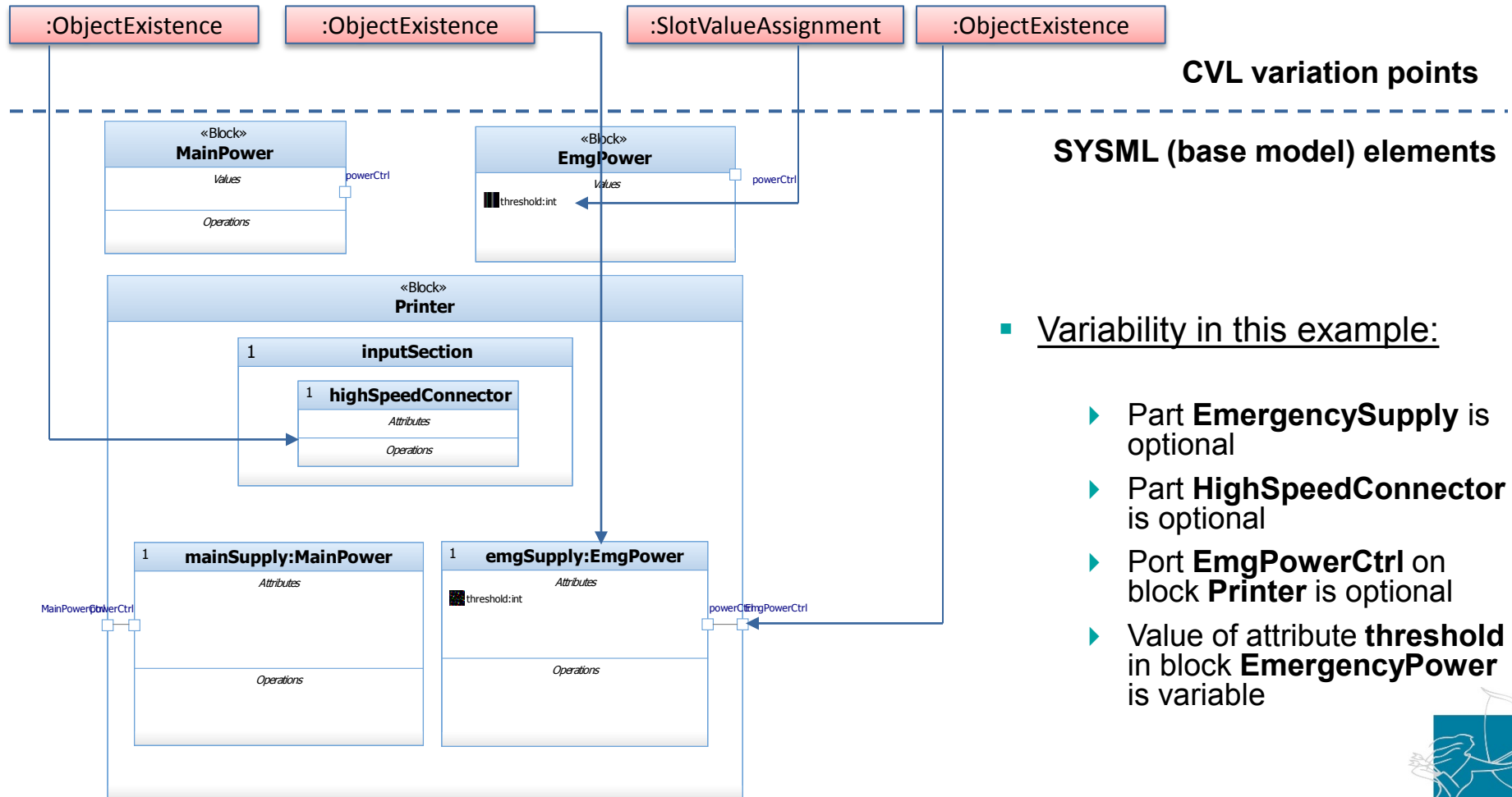


Common Variability Language (CVL)





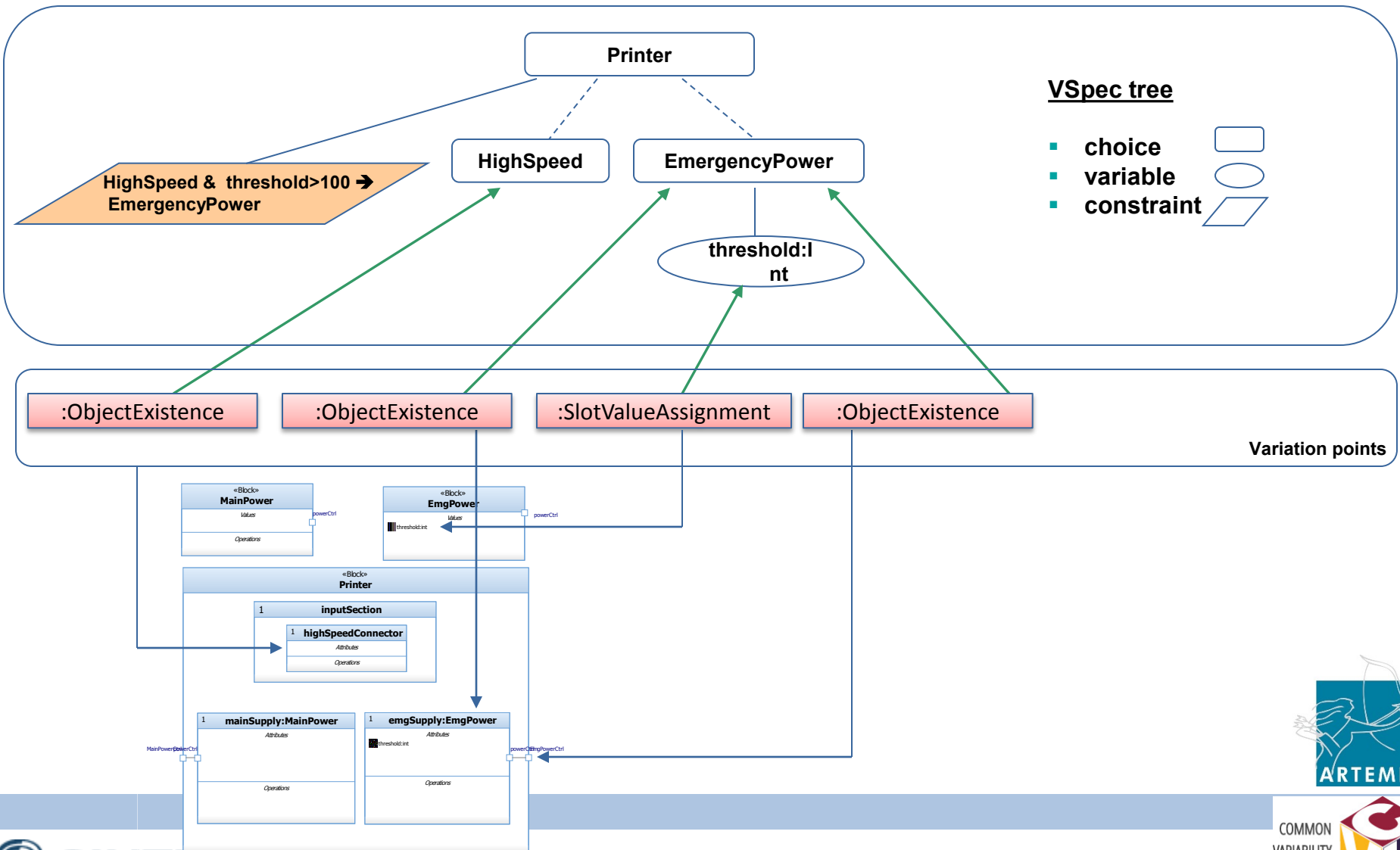
Variation Points over base model



- Variation Points refer to Base objects
- Variation Points define the base model modifications precisely
- There are different kinds of Variation Points
 - Existence
 - Value assignment
 - Substitution
 - Opaque variation point
 - Configurable Unit



VSpec trees and binding

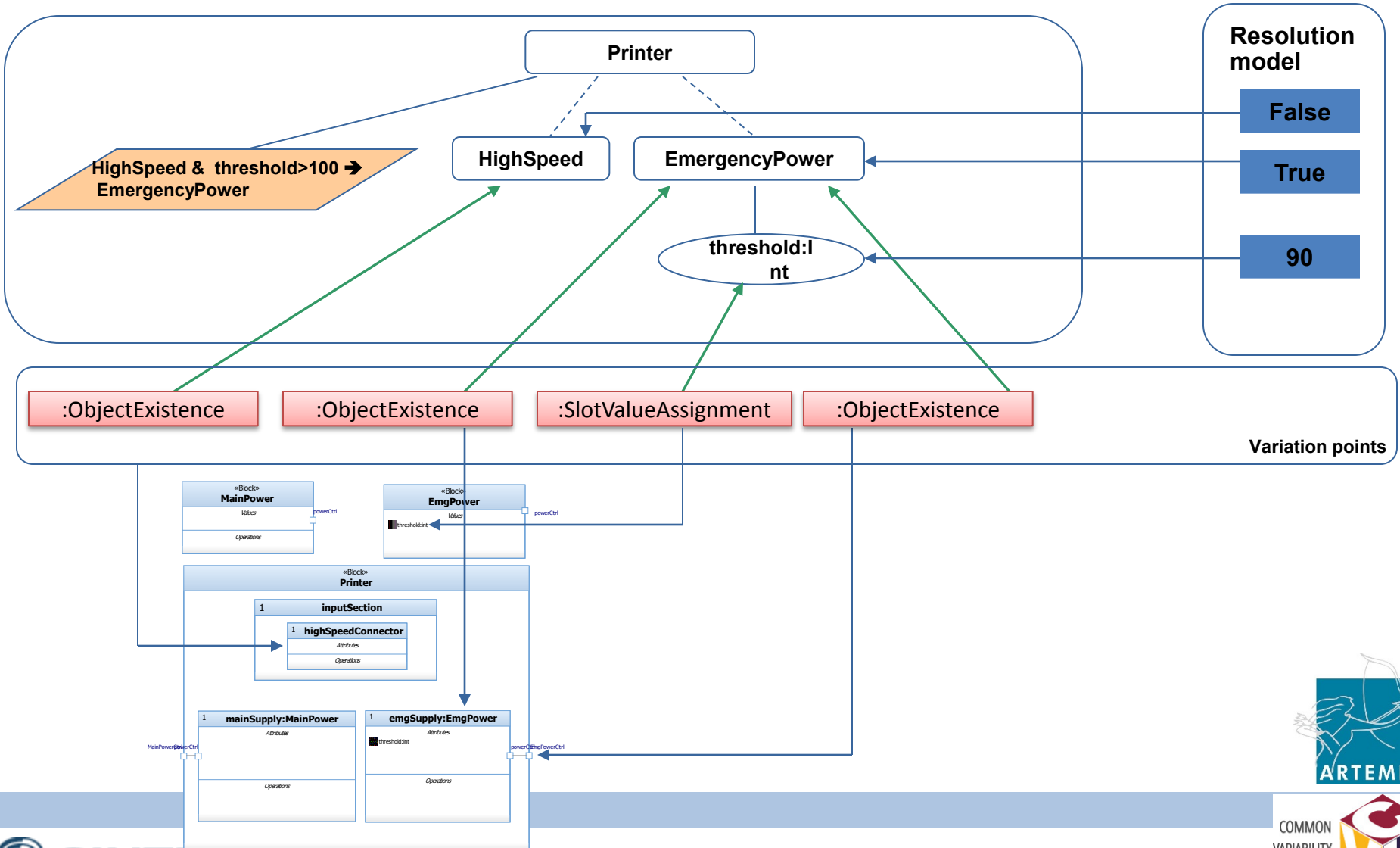


- VSspecs (Variation Specifications) describe the abstract variability
- Every Variation Point is bound to exactly one VSspec
- VSspecs come in different kinds:
 - Choice
 - Variable
 - Constraint
 - VClassifier
 - CVSpec



- CVL include a basic language for expressing constraints on the VSpec tree
 - Propositional logic is supported
- CVL also has the opportunity to let you apply other constraint languages like OCL

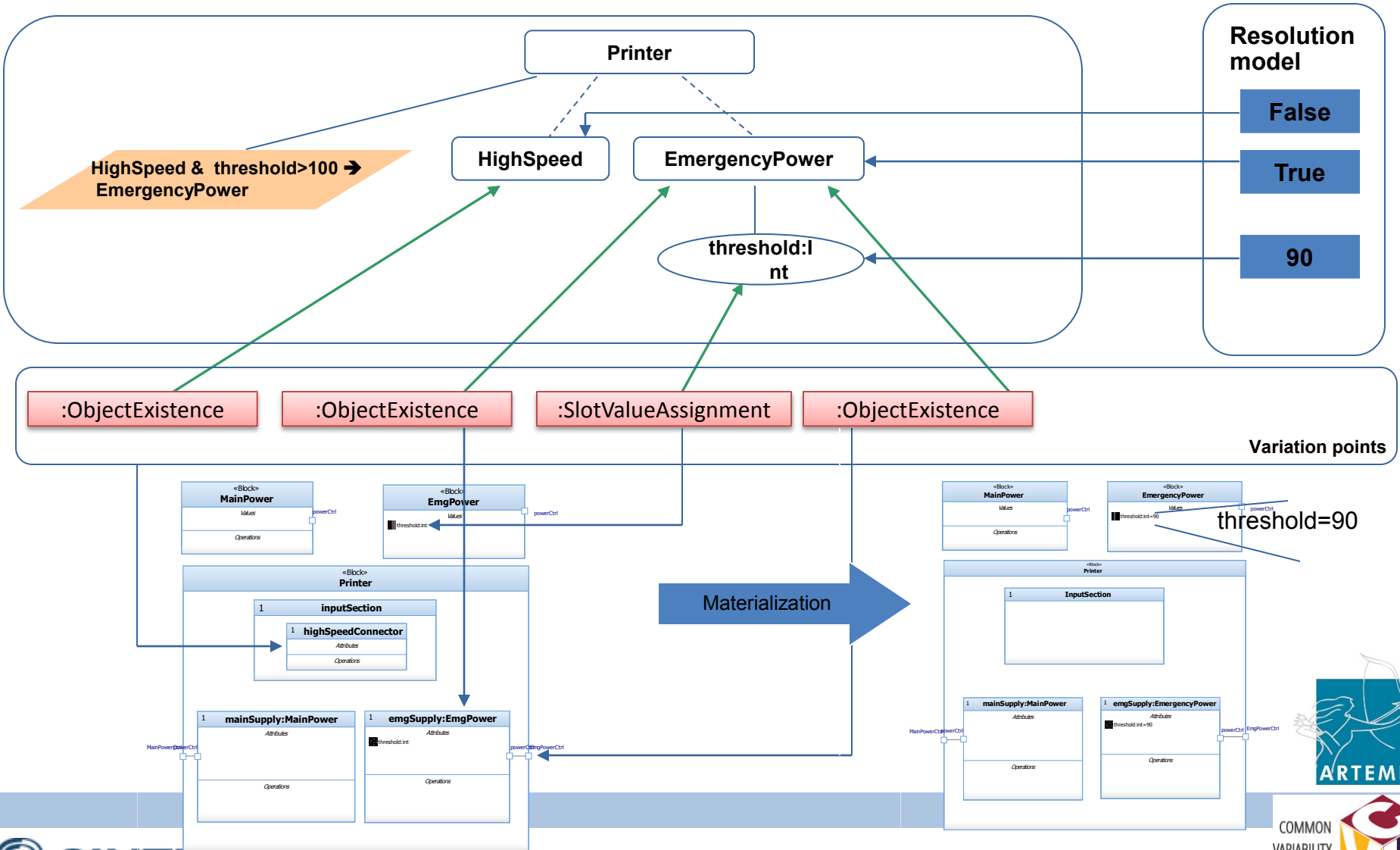




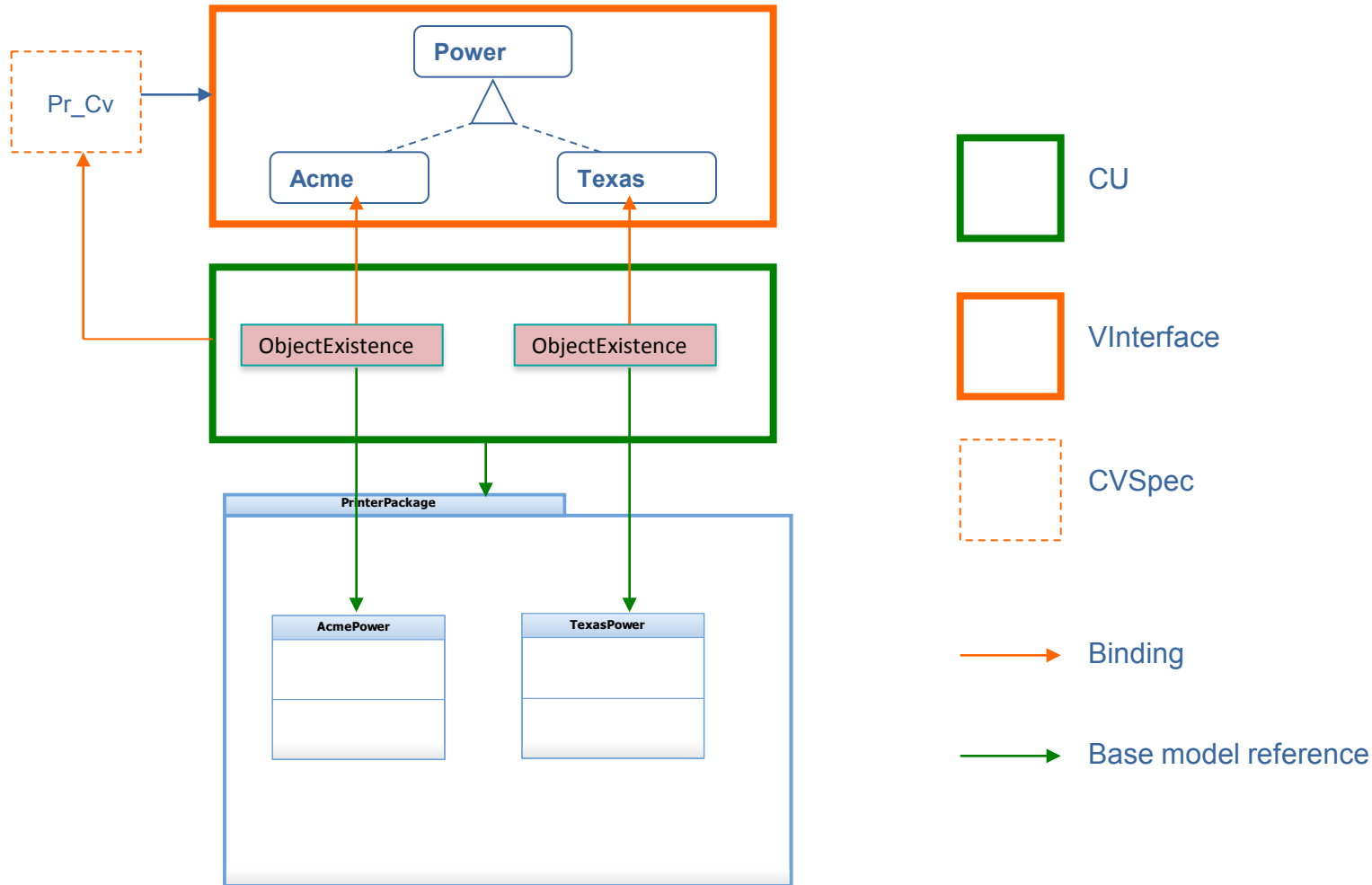
- VSpecResolution elements refer to VSpecs
- The set of valid Resolutions is restricted by the constraints
- Represent information necessary to materialize product models
 - Actual yes/no decisions on Choices
 - Actual values to Variables
 - Instances of VClassifiers
 - Configurations of CVSpecs/Configurable Units



Materialization



Configurable Unit and VInterface



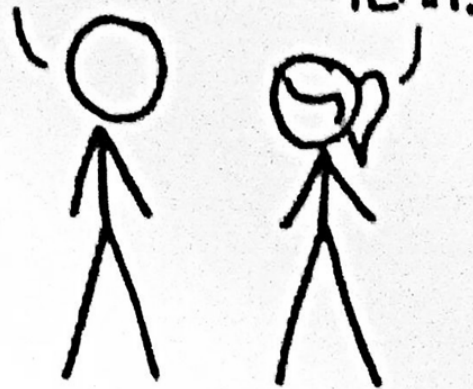
- Please look at <http://variabilitymodeling.org>
- There you will find e.g.
 - The Revised Submission of CVL
 - Tutorial slides from SPLC tutorial on CVL
 - Links to supplementary, and historical material



HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.



SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.





VARIES



Why standardize a language?



- Common terms and interpretation
 - across persons, teams, companies and cultures
 - Experience SISU project
 - Very large SDL specification ported from Alcatel to Kongsberg
 - Experience MSC
 - We have a Korean translation of MSC 2000
 - across computers! portability
 - Experience Simula
 - We ported the exact same code on at least 5 machines without changing a single line of code around 1980
- Common teaching material
- Common libraries
- Common and open reviewing process





"OMG is valuable from many different perspectives: For the small tool vendors, OMG may provide standards that let the small SMEs play on the same field as the big ones through interfacing through the same technology. For the large industry, OMG may provide means to establish the fundamentals of multiple vendors of support tooling. For the academic, OMG is a challenge because you have to defend your ideas in a larger context, and for the industrial researcher, OMG is a vehicle for disseminating industrial results to a wider community in a form that many companies can attach to. "

-Dr. Øystein Haugen, Senior Researcher, SINTEF

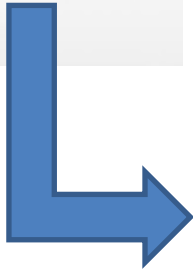
- OMG is valuable from many different perspectives:
 - For the small tool vendors,
 - OMG may provide standards that let the small SMEs play on the same field as the big ones through interfacing through the same technology.
 - For the large industry,
 - OMG may provide means to establish the fundamentals of multiple vendors of tooling.
 - For the academic,
 - OMG is a challenge because you have to defend your ideas in a larger context, and
 - for the industrial researcher,
 - OMG is a vehicle for disseminating industrial results to a wider community in a form that many companies can attach to.



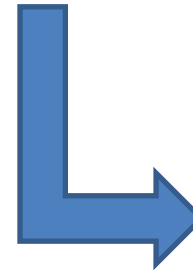
Why make something special/proprietary? VARIES



Nokia N95



iPhone



HTC Wildfire





JOINT SUBMISSION TEAM

initiators from the MoSiS project



tool vendors



users & consultants



research institutes and universities



Vanity? Why should we succeed?



- In Variability Modeling there has been
 - focus on notation
 - focus on the abstraction layer (feature models)
 - few attempts to standardize formally
- but for some reason now things start to happen
 - Separate language (variability language as such)
 - CVL (our baby)
 - Amalgamated approaches (variability combined into another language)
 - AUTOSAR
 - Matlab Simulink
 - SysML



Success is not guaranteed



- Risk 0: Nobody cares about your proposed standard
 - This is why I prepared OMG over many years (since 2005)
- Risk 1: You meet strong opposition
 - This once happened when we wanted OO into SDL (1989-1992)
 - This happened again when we wanted UML more precise (1999)
- Risk 2: You run out of time and money
 - Research projects are seldom more than 4 years
 - Standardization may easily take longer
- Risk 3: You achieve a standard, but nobody makes tools
 - This has happened to many standards
- Risk 4: There is a standard, there are tools, but no users
 - May be your compromises were just not good enough



- CVL emerges from a series of European research projects:
 - CAFE, Families, MoSiS, VERDE, CESAR, VARIES, ...
- CVL has a true international consortium
 - Europe, North-America, India
- CVL has a fine mixture of competence
 - Tool vendors (Atego, pure-systems, IBM)
 - Serious users (Thales, TCS)
 - Reknowned scientists from research institute and universities



- For the academic:
 - Cannot invent new things all the time
 - Needs to have the totality in mind
 - not only a very limited part of the domain/language
 - no assumptions of simplification
 - Little personal credit for authoring a standard
- For the industrial researcher
 - The tool prototype is lagging behind the language
 - CVL 1 Tool is supporting something similar to OMG CVL but ...
 - ATL and MOFScript were not quite supporting the OMG standard
 - The project terminates with no standard in sight



Limitations? (2)



- For the tool vendor:
 - The compromises of the standardization increases the need to modify the tool they already have
 - They always want the simplest solution that they believe their customers are asking for
 - The solution perceived as simplest may also be the most special
- For the industrial early adopter:
 - Tool support is lacking!
 - As an Ericsson representative I experienced that with SDL, MSC and UML
 - The more we are asking for of the standard, the more the tooling is lagging



Limitations? (3)



US007543269B2

(12) United States Patent Krueger et al.

(10) Patent No.: **US 7,543,269 B2**
(45) Date of Patent: **Jun. 2, 2009**

(54) SOFTWARE CUSTOMIZATION SYSTEM AND METHOD

(75) Inventors: **Charles W. Krueger**, Austin, TX (US);
Mark N. Jungman, Cedar Park, TX (US)

(73) Assignee: **BigLever Software, Inc.**, Austin, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 605 days.

(21) Appl. No.: **10/106,154**

(22) Filed: **Mar. 26, 2002**

(65) **Prior Publication Data**
US 2002/0154555 A1 Oct. 24, 2002

Related U.S. Application Data
(60) Provisional application No. 60/278,786, filed on Mar. 26, 2001.

(51) Int. Cl. **G06F 9/44** (2006.01)
(52) U.S. Cl. **717/107; 717/102; 717/105; 717/108; 717/109**

(58) **Field of Classification Search** **717/106, 717/104, 100, 108, 122**
See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

6,016,394 A * 1/2000 Walker 717/104
6,789,252 B1 * 9/2004 Burke et al. 717/100
6,851,105 B1 * 2/2005 Coad et al. 717/106
6,993,743 B2 * 1/2006 Crupi et al. 717/102

OTHER PUBLICATIONS

Jan Bosch, "Design and Use of Software Architecture: Adopting and Evolving a Product-Line Approach", May 2000, Addison-Wesley, First Edition, Chapter 7, pp. 181-196.*
Charles W. Krueger, "Modeling and Simulating a Software Architecture Design Space", 210 pages (38 pages extracted). [Online] [Retrieved at] <reports-archive.adm.cs.cmu.edu/anon/1997/CMU-CS-97-158.pdf>.*

Sholom Cohen, "Guidelines for Developing a Product Line Concept of Operations", Aug. 1999, 77 pages. [Online] [Retrieved at] <www.sei.cmu.edu/pub/documents/99/reports/pdf/99tr008.pdf>.*
"Software Product Line Reuse in Practice", C. Krueger, Dec. 2000. Online retrieved at <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=60888062>.*

"The First Software Product Line Conference—Technical Papers", Aug. 2000. Online retrieved at <http://www.sei.cmu.edu/plp/conf/SPLCL_authors_and_abstracts.html>.*
"Transitioning Legacy Assets To A Product Line Architecture", Bayer et al., Dec. 1999. Online retrieved at <http://delivery.acm.org/10.1145/320000/>.*

* cited by examiner

Primary Examiner—Tuan Q Dam

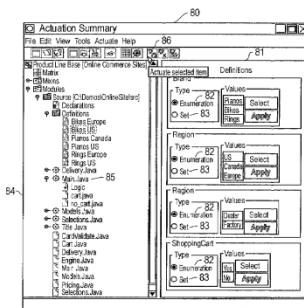
Assistant Examiner—Thuy Dao

(74) **Attorney, Agent, or Firm**—DuBois, Bryant & Campbell, LLP; William D. Wiese

(57) ABSTRACT

A system and method for the mass customization of software includes a software production line infrastructure, development environment, and actuator. The infrastructure organizes the software production line using a feature declarations component, a product definitions component, an automaton component, and a partition composition component. The development environment browses, displays, organizes, edits, and maintains the infrastructure. The actuator actuates the software production line to produce custom software products.

17 Claims, 10 Drawing Sheets



- BigLever (Krueger) has one US patent and a couple of patent applications
- Krueger claims that any tool that supports CVL may infringe upon these patents
- Krueger does not want to be more specific than this and says that the information is in the patent papers



- **US Patent, US 7,543,260 B2, Issued June 2, 2009.** "Software Mass Customization System and Method"
 - *Abstract:* A system and method for the mass customization of software includes a software production line infrastructure, development environment, and actuator. The infrastructure organizes the software production line using a feature declarations component, a product definitions component, and automaton component and a partition composition component. The development environment browses, displays, organizes, edits, and maintains the infrastructure. The actuator actuates the software production line to produce custom software products.
- **US Patent Application, 12/273,352, Filed Nov 18, 2008.** This contains claims that are still pending from the original patent listed above.
- **US Patent Application, 12/115,616, Filed May 6, 2008** (Provisional file date May 7, 2007). "Model Elements in Software Tools as Automaton in a Software Customization System and Method."



What happens now with CVL?



Date: August 13, 2012



Common Variability Language (CVL)

OMG Revised Submission

OMG document: ad/2012-08-05

Submitters

IBM
Fraunhofer FOKUS
Thales
Tata Consultancy Services

Supporters

SINTEF
University of Oslo
Tecnalia Research & Innovation
University of Waterloo
IT University of Copenhagen
INRIA
CEA
Atego
Pure-systems

Primary Contact:

Øystein Haugen, SINTEF
oystein.haugen@sintef.no



- CVL Revised Submission exists
- CVL was presented to the OMG Architectural Board in December 2012
 - The AB will vote on adoption by e-mail once some technicalities of the metamodel has been cleared
- When technically adopted the FTF phase starts
 - FTF = Finalization Task Force
 - Tool vendors will implement CVL tooling and find problems
- There will also be a vote by the OMG Business Committee on the business aspects of the standard.
 - They will also consider the patent infringement possibilities
- After successful FTF then CVL will be "available technology"



Potential tool vendors for CVL?



- Research/experimental tools
 - from SINTEF, INRIA
- Internal proprietary tools
 - potentially from TCS
- Commercial tool vendors in consortium
 - IBM, pure-systems, Atego
- Commercial tool vendors outside consortium
 - Big Lever, NoMagic, Sparx,
- Where to find the CVL Revised Submission?
 - <http://variabilitymodeling.org/>



- Common or Chaos?
- Variability Language or a Variety of languages?
- Limitations or leverage of the community?

